UDC 004.932:528.854
DOI https://doi.org/10.32782/IT/2024-3-13

Leonid MESHCHERIAKOV
Doctor of Technical Sciences, Professor, Professor at the Department of Software Engineering, Dnipro University of Technology, 19, Dmytra Yavornytskoho Ave., Dnipro, Ukraine, 49005, meshcheriakov.l.i@nmu.one
ORCID: *0000-0002-9579-19701970*
Scopus-Author ID: *57205282540*

Nataliia ULANOVA
Candidate of Technical Sciences, Associate Professor at the Department of Applied Mathematics, Dnipro University of Technology, 19, Dmytra Yavornytskoho Ave., Dnipro, Ukraine, 49005, ulanova.n.p@nmu.one
ORCID: *0000-0001-8460-5266*

Vira PRYKHODKO
Candidate of Technical Sciences, Associate Professor at the Department of Applied Mathematics, Dnipro University of Technology, 19, Dmytra Yavornytskoho Ave., Dnipro, Ukraine, 49005, prykhodko.v.v@nmu.one
ORCID: *0000-0002-5669-5927*

Mykhailo PIMAKHOV
Bachelor at the Department of Computer System's Software, Dnipro University of Technology, 19, Dmytra Yavornytskoho Ave., Dnipro, Ukraine, 49005, pimakhov.my.v@nmu.one

To cite of article: Meshcheriakov, L., Ulanova, N., Prykhodko, V., Pimakhov, M. (2024). Prohramuvannia protsesiv keruvannia ta lohiky na stseni virtualnoi audytorii v seredovyshchi ArchViz [Programming control processes and logic on the stage of a virtual audience in the ArchViz environment]. *Information Technology: Computer Science, Software Engineering and Cyber Security*, 3, 124–132, doi: https://doi.org/10.32782/IT/2024-3-13

PROGRAMMING CONTROL PROCESSES AND LOGIC ON THE STAGE OF A VIRTUAL AUDIENCE IN THE ARCHVIZ ENVIRONMENT

The scope of application of virtual models of the final product has great potential for presentation to consumers. By offering an interactive exploration of the virtual classroom space, users can interact with and manipulate various elements of the environment, gaining a deeper understanding of the space and its potential use.

The aim *of the work is to present the process of practical implementation of the creation of an application of a virtual environment for architectural visualization of a university classroom using the Blueprint programming system.*

The methodology *for ensuring the application of the development of the interactive application ArchViz, which allows you to explore the created virtual audience with the possibility of interactive interaction with all its main elements.*

The scientific novelty *of the proposed solutions is determined by the fact that through the use of advanced features of Unreal Engine 5, such as a dynamic lighting system, real-time global lighting, and high-precision rendering capabilities, the ArchViz application achieves a new increased level of realism and interactivity in visualizing the virtual environment.*

Conclusions. *The practical value of the study lies in the ability to comprehensively consider interior design options through virtual models without the need for physical reconstruction or presence in real space.*

Key words: *virtual realism, ArchViz, Blueprint, Niagara.*

Леонід МЕЩЕРЯКОВ
доктор технічних наук, професор, професор кафедри програмного забезпечення комп'ютерних систем, Національний технічний університет «Дніпровська політехніка», просп. Дмитра Яворницького, 19, м. Дніпро, Україна, 49005
ORCID: *0000-0002-9579-1970*
Scopus-Author ID: *57205282540*

Наталія УЛАНОВА
кандидат технічних наук, доцент кафедри прикладної математики, Національний технічний університет «Дніпровська політехніка», просп. Дмитра Яворницького, 19, м. Дніпро, Україна, 49005
***ORCID:** 0000-0001-8460-5266*

Віра ПРИХОДЬКО
кандидат технічних наук, доцент кафедри прикладної математики, Національний технічний університет «Дніпровська політехніка», просп. Дмитра Яворницького, 19, м. Дніпро, Україна, 49005
***ORCID:** 0000-0002-5669-5927*

Михайло ПІМАХОВ
бакалавр кафедри програмного забезпечення комп'ютерних систем, Національний технічний університет «Дніпровська політехніка», просп. Дмитра Яворницького, 19, м. Дніпро, Україна, 49005

Бібліографічний опис статті: Мещеряков, Л., Уланова, Н., Приходько, В., Пімахов, М. (2024). Програмування процесів керування та логіки на сцені віртуальної аудиторії в середовищі ArchViz. *Information Technology: Computer Science, Software Engineering and Cyber Security*, 3, 124–132, doi: https://doi.org/10.32782/IT/2024-3-13

ПРОГРАМУВАННЯ ПРОЦЕСІВ КЕРУВАННЯ ТА ЛОГІКИ НА СЦЕНІ ВІРТУАЛЬНОЇ АУДИТОРІЇ В СЕРЕДОВИЩІ ARCHVIZ

Область застосування віртуальних моделей кінцевого продукту має в майбутнім великий потенціал для представлення споживачам. Пропонуючи інтерактивне дослідження простору віртуальної аудиторії, користувачі можуть взаємодіяти з різними елементами середовища та маніпулювати ними, отримуючи глибше розуміння простору та його потенційного використання.

***Метою роботи** є представлення процесу практичної реалізації створення додатку віртуального середовища архітектурної візуалізації аудиторії університету за допомогою системи програмування Blueprint.*

***Методологія** забезпечення застосування розробки інтерактивного додатку ArchViz, що дозволяє досліджувати створену віртуальну аудиторію із можливістю інтерактивної взаємодії з всіма основними її елементами.*

***Наукова новизна** запропонованих рішень визначається тим, що завдяки використанню передових можливостей Unreal Engine 5, таких як система динамічного освітлення, глобальне освітлення в реальному часі та можливості високоточного рендерингу, додаток ArchViz досягає нового підвищеного рівня реалістичності та інтерактивності візуалізації віртуального середовища.*

***Висновки.** Практична цінність дослідження полягає в можливості через віртуальні моделі здійснити всебічний огляд варіантів дизайну приміщень без необхідності фізичної реконструкції або присутності в реальному просторі.*

***Ключові слова:** віртуальний реалізм, ArchViz, Blueprint, Niagara.*

Актуальність проблеми. Прогнози щодо розвитку досліджень віртуального моделювання вказують на подальший прогрес у сфері додатків в середовищі *ArchViz.* Основною метою тут являється встановлення нового стандарту якості, використовуючи досягнення в області технологій, фотореальної графіки та інтерактивності, що надаються рушієм *Unreal Engine 5*. Розсуваючи межі можливого, очікується, що майбутні розробки ще більше підвищать реалістичність, занурення та зручність використання додатків *ArchViz*, встановлюючи нові стандарти для індустрії віртуального моделювання.

Аналіз останніх досліджень і публікацій. Методи створення інтерактивних *3D*-презентацій, зокрема у сфері архітектурної візуалізації, з використанням рушія *Unreal Engine 5* на даний час знаходять все більше застосування (Everett Gunther, 2022; Unreal Sensei, 2022). Причому самі ігрові рушії визначаються як програмний фреймворк або платформа, яка надає розробникам набір інструментів, бібліотек та систем для створення, розробки і розгортання відеоігор, що слугує проміжною ланкою між кодом гри та апаратним забезпеченням, дозволяючи розробникам зосередитися на логіки та дизайні гри. Із популярних ігрових рушіїв можна виділити перед усе такі як *Unity, Unreal Engine* та *CryEngine*. *Unity* широко використовуваний ігровий рушій, відомий своєю універсальністю та простотою використання. *Unreal Engine* відомий своїми дуже широкими візуальними можливостями та високою точністю комп'ютерної графіки. Він

надає такі розширені можливості як трасування променів світла у реальному часі та динамічне глобальне освітлення, що дозволяє розробникам створювати візуально вражаючі ефекти. *CryEngine* в основному зосереджений на створенні найсучаснішої комп'ютерної графіки та реалістичних середовищ. Серед розглянутих, найбільш широко використовуваних двигунів, для розробки *ArchViz*-проекту найкращим виступає *Unreal Engine 5*, як оптимальний за багатьма параметрами з усіх запропонованих, що містить найновітніші технології та має можливість програмувати за допомогою системи *Blueprint*.

Метою статті є представлення процесу практичної реалізації створення додатку віртуального середовища архітектурної візуалізації аудиторії університету за допомогою системи програмування *Blueprint*.

Виклад основного матеріалу. Для програмування взаємодії з оточенням на сцені віртуальної аудиторії було розроблено візуальний інтерфейс через клас *UserWidget*. Наверху по центру було вирішено розташувати годинник та слайдер для зміни часу. Слайдеру задано значення 960 (значення співпадає зі стартовим значенням часу в об'єкті *Ultra_Dynamic_Sky*) та величину кроку виставлено 0.000001 (Pamir Garay, 2022; Shoun Foster, 2023), щоб точність зміни часу залежала лише від роздільної здатності дисплею користувача. В якості годинника у віджет *LevelWidget* інтегровано віджет *UDS_Digital_Clock*, що йде у комплекті з *Ultra_Dynamic_Sky* та має пряму прив'язку до нього по параметру часу.

В правому верхньому куті розташовано панель для керування погодою. Ця панель складається з елементів *Slider*, *Image* та скопійовано елементи *Background* та *Background Blur* з віджету *UDS_Digital_Clock* для створення єдиного стилю. Після створення та розташування елементів, до кожного із них було застосовано прив'язку *Anchor*, аби незалежно від розміру вікна та роздільної здатності екрану, усі користувачі мали однакове розташування елементів інтерфейсу.

Після розробки візуальної складової користувацького інтерфейсу запрограмовані його елементи для досягнення бажаного функціоналу. У вкладці *Event Graph* до події *Event Construct* підключено блоки з отриманням інформації про об'єкти *Ultra_Dynamic_Sky* та *Ultra_Dynamic_Weather* та задано цим об'єктам змінні-референси (рис. 1) (PrismaticaDev, 2023; Adam the Chips, 2022).

Елементу *Slider_0* потрібно задати функціонал для зміни часу доби. Для цього, в розділі *Event Graph* обрано цей елемент та додано подію *On Value Changed*, що відповідає за те, що відбудеться при зміні значення даного елементу (рис. 2). У вхід *Set Time Of Day* прив'язано значення із плаваючою точкою *Value* із події *On Value Changed (Slider_0)*, а у вхід *exec* прив'язано вихід *exec* з блоку *Get All Actors Of Class*. Таким чином, логіка зміни часу користувачем буде така: якщо змінено значення слайдеру → відтворити зміну параметру часу на встановлене значення слайдеру для отриманого раніше об'єкту *Ultra_Dynamic_Sky* (референс до змінної *Time of Day*).

Після програмування зміни часу доби, потрібно запрограмувати зміну погоди. Для того, щоб користувач міг змінити погоду на потрібну необхідно додати зображення,

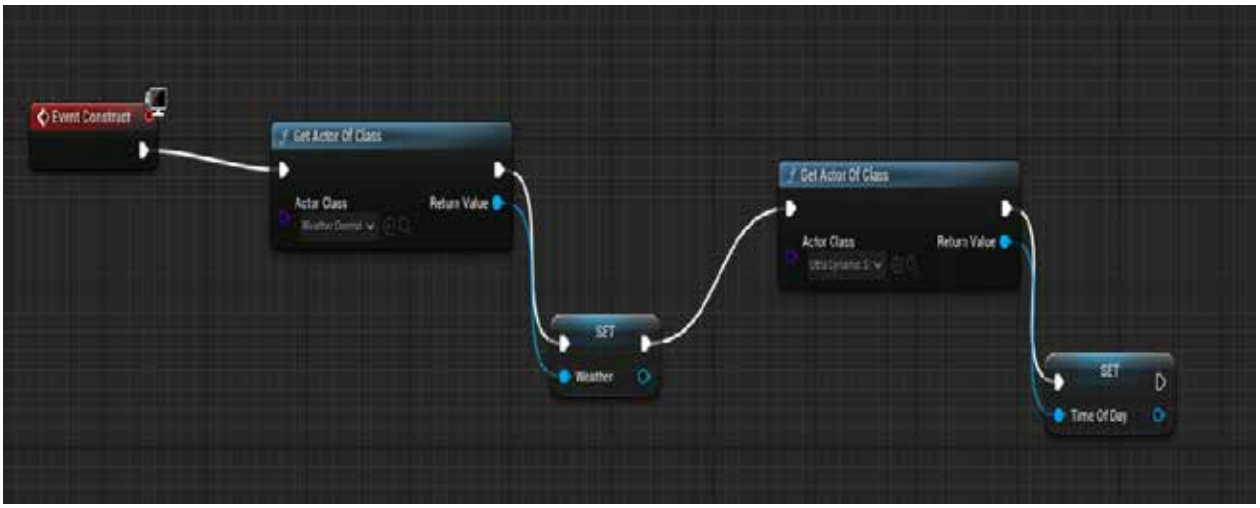


Рис. 1. Логіка *Ultra_Dynamic_Sky* та *Ultra_Dynamic_Weather*

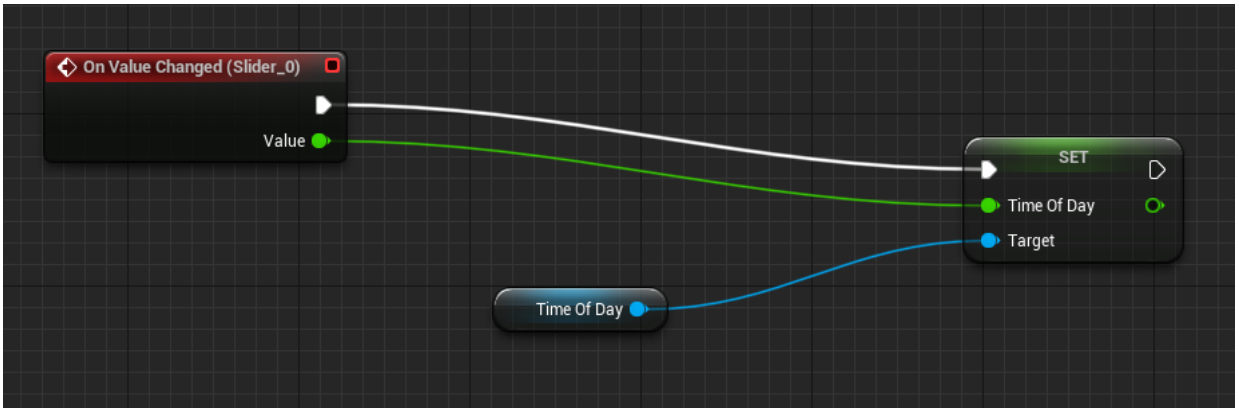


Рис. 2. Логіка для задання часу

що відповідатимуть заданому параметровому стану погоди. Зображення ідентифікації погоди залучено з сервісу *iconfinder.com*, відредаговано та змінено їх кольори на білий та створено з них додаткового зображення *partly_cloudy_weather_icon.png*. Усього використовується сім станів погоди: сонячна *(Clear_Skies)*, хмарна *(Cloudy)*, похмуро *(Overcast)*, частково хмарна *(Partly_Cloudy)*, дощ *(Rain)*, легкий дощ *(Rain_Light)* та дощ із грозою *(Rain_Thunderstorm)*. Кожному з цих станів відповідає своє зображення.

Після цього, в *Event Graph*, подібно до часу доби, але вже для елементу *Slider_1* створено подію *On Value Changed* та створено логічний ланцюг з блоків, логіка якого така: якщо змінено значення слайдеру то перетворити це значення в ціле число, а якщо це значення дорівнює порядковому номеру погоди на слайдері (*0 – 6*), то для отриманого раніше об'єкту задати відповідне значення погоди згідно його порядку та встановити відповідну піктограму в елемент *Image_0*.

Також, для інформування користувача про клавіші для керування оточенням, було створено окремий віджет з назвою *ControlsWidget*, що містить у собі напівпрозорий елемент *Backgound* та розмиття *Background Blur*.

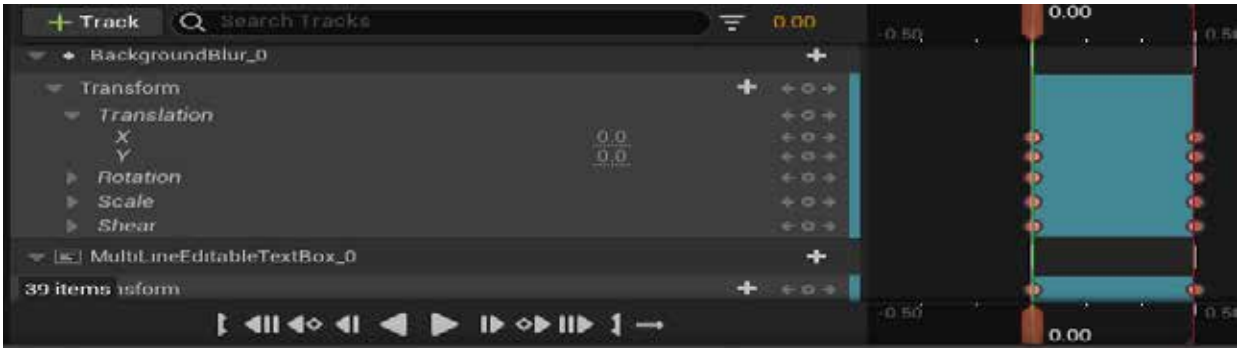
Після створення анімації шляхом проставлення кейфреймів та логіки віджету, досягнуто функціоналу, що дозволяє показати та приховати цей віджет із супроводженням плавної анімації (рис. 3).

Логіка сцени. На сцені віртуальної аудиторії присутні два об'єкти що мають містити в собі логіку для взаємодії з оточенням, а саме: настінний годинник та ноутбук. Ці два об'єкти повинні показувати поточний час доби на сцені в режимі реального часу із врахуванням можливості контролю часу користувачем. Так як проект вже містить в собі систему глобального освітлення, а саме *Ultra Dynamic Sky*, реалізація цього значно спрощується. Для того, щоб ноутбук мав відображення поточного часу, було створено віджет *PCTime*, який містить у собі лише елемент *Image* із скріншотом *OC Windows 11* та заздалегідь дубльований віджет *UDS_Digital_Clock*, але без фону та заданим чорним кольором шрифту.

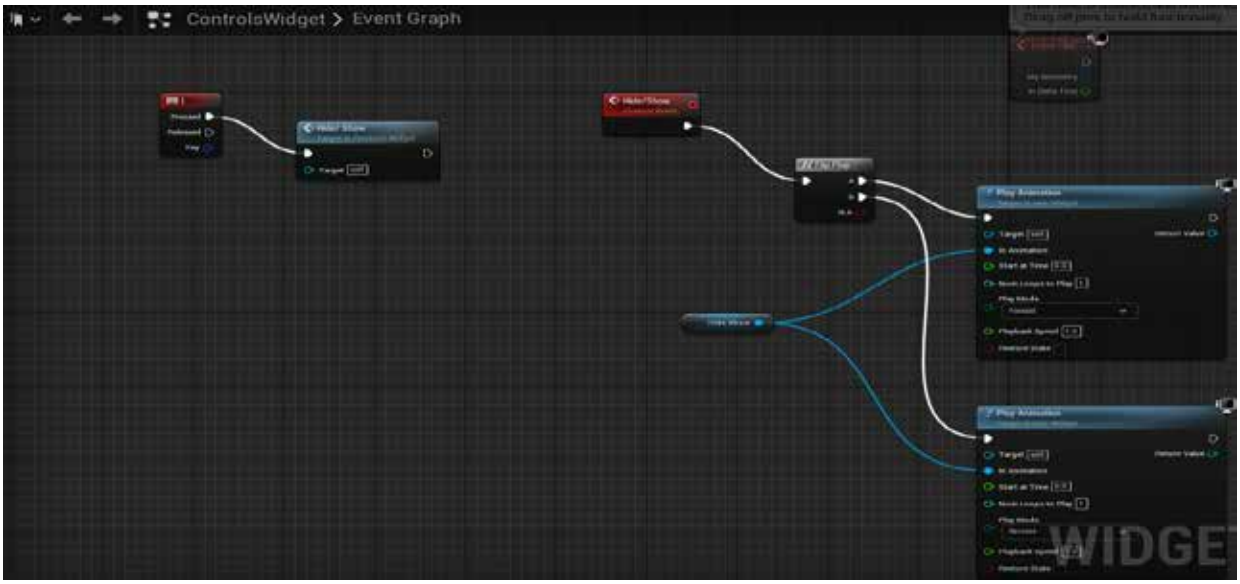
В подальшому створено об'єкт *PCTimeActor* класу *Actor* через розділ *Blueprint Class* та додано до нього елемент *Widget*, що посилається на *PCTime*. Це зроблено для можливості розміщення цього віджету у *3D*-просторі (рис. 4).

Після створення цього об'єкту, його було розміщено на координатах екрану ноутбуку. Таким чином отримано імітацію робочого ноутбуку через показання на ньому реального часу. А для того, щоб настінний годинник також показував поточний час, зі сцени було вилучено меші (об'єкти *Static Mesh*) годинникових стрілок та створено об'єкт *WallClock*, що містив у собі ці меші. Після цього годинникові стрілки було запрограмоване таким чином, щоб вони брали поточний час з компоненту *Ultra Dynamic Sky*, для часової стрілки бралося 1200%, а для хвилинної – 100% і здобуті значення застосовувались до власного обертання. Увесь цей код виконується кожний тік, що дозволяє стрілкам обертатися плавно, як це відбувається в реальному житті.

Також, на сцені містяться об'єкти з якими користувач може взаємодіяти самостійно. Серед них: лампи на стелі, підлога, крісла в лаундж-зоні та диван з піддонів, що початково знаходиться поза межами аудиторії. Кожен з цих видів об'єктів було запрограмовані. Так для ламп був написаний код з подіями, при запуску яких відбувається зміна значення параметру у *LampMatParameter* між станами вимкнено (значення *0.0*) та ввімкнене (значення *1.0*). Для



а.



б.

Рис. 3. Анімація та логіка *ControlsWidget*:
а. – анімація *ControlsWidget*; б. – логіка *ControlsWidget*

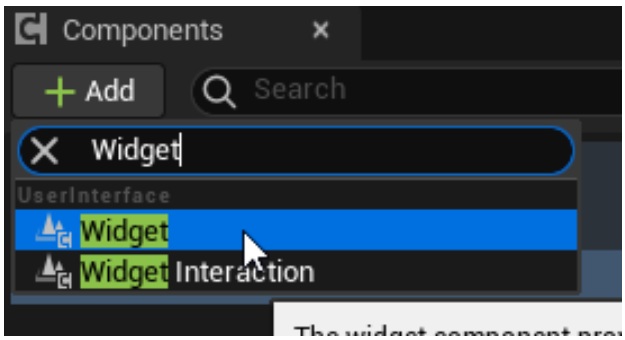


Рис. 4. Створення *PCTimeActor*, що містить в собі *PCTime*

захисту від випадкової взаємодії з лампами (взаємодія відбувається шляхом натиснення клавіші *E*) було вирішено поставити порядок блоку *Flip Flop* таким чином, щоб при першому натисненні на клавішу, до ламп застосовувалося значення «вимкнено». Саме тому після ввімкнення додатку, для найпершої взаємодії з лампами, клавішу потрібно буде натиснути двічі.

Інші об’єкти перед взаємодією при наведенні на них курсором миші, повинні мати обведення, для того аби виділятись на фоні інших об’єктів, що означає, що користувач збирається взаємодіяти саме з цим об’єктом. Також це обведення допоможе користувачу зрозуміти, які об’єкти на сцені взагалі є інтерактивними та з якими можна взаємодіяти. Для цього зі стандартного шаблону *Unreal Engine 4* було імпортовано матеріал для пост-процесінгу під назвою *M_Highlight* та додано його до *PostProcessVolume*.

Далі, для підлоги та стандартних крісел було застосовано один і той самий код з логікою, що запускається кожен тік: визначення стану користувача (чи знаходиться він в режимі редагування) → якщо так, то застосувати до об’єкту параметр глибини (для отримання обведення)

та якщо користувач кликне по об'єкту, з'явиться інтерфейс для зміни кольору, а якщо до цього було відкрито інтерфейс для редагування іншого об'єкту, то він закривається, а обведення того об'єкту припиняється, як при відведенні з нього миші або при виході з режиму редагування. Цей код міститься в кожному із вищезазначених об'єктів, але з різною варіативністю деяких змінних.

Також, для кожного з цих об'єктів було створено власний віджет-інтерфейс взаємодії. Але, існують три об'єкти, в яких є додатковий функціонал. З них – два крісла та один диван. Їх особливість полягає в тому, що при взаємодії з ними, відкривається інтерфейс, що окрім звичайного функціоналу зміни кольору містить в собі кнопку для взаємозаміни (крісла замінюються диваном, а диван – кріслами). Програмно віджети для взаємодії з цими об'єктами відрізняються наявністю виклика події заміни, яку містить у собі диван. Програмна частина спеціальних крісел ніяк не відрізняється від інших.

У дивану код такий самий, як і в інших об'єктів за винятком однієї особливості: він містить у собі анімації переміщення крісел за межі сцени й анімацію переміщення в аудиторію. Анімація є поступовою (показує, як збирається диван), унаслідок чого він не є одним об'єктом – це 5 різних об'єктів, пов'язаних між собою лише інтерфейсом зміни кольору й одночасним завданням обведення для всього дивана, щоб створити видимість цілісності виробу. Анімація зроблена переміщенням по заданим координатам за даний відрізок часу. Усі анімації переміщень мають тривалість 2 секунди, тож для всіх них використовується однаковий блок *Timeline*. Таким чином, було отримано можливість інтерактивної взаємодії з різними аспектами віртуального середовища: від заміни кольору підлоги та меблів до заміни їх типу.

Програмування керування. Для того, щоб користувач міг взаємодіяти зі створеним *User Interface*, перш за все, його потрібно вивести у робочу область дисплею. Для цього у *BP_FirstPersonCharacter* потрібно при запуску проекту (подія *Event BeginPlay*) створити ці віджети та додати їх до *Viewport* (рис. 5).

Після того, як інтерфейс був виведений на екран, потрібно додати гарячу клавішу *P* для того, щоб вмикався режим редагування (з'являвся курсор миші, який вже може взаємодіяти із віджетами та об'єктами, в налаштуваннях яких включена функція *Generate OnClick Events*) (рис. 6).

Також, користувачеві потрібно надати можливість наближувати картинку клавишею *C.* Принцип роботи: при затисненні клавіші *C* відбувається плавна анімація зменшення кута поля зору вдвічі, а при відпусканні клавіші *C* – програється зворотна анімація. Також, за допомогою глобальної змінної *Is Editing* було запрограмоване, щоб користувач не міг збільшувати зображення під час редагування. Так як було створено матеріал та набір параметрів для вмикання/вимикання штучного світла від ламп на стелі, потрібно задати гарячу клавішу *E* на цей функціонал.

Покращення візуального та звукового сприйняття додатку. Щоб покращити сприйняття додатку користувачем, було вирішено додати деякі елементи до проекту, а саме фонову музику, створену нейромережею *soundraw.io*. Спочатку було сформовано музикальну чергу із чотирьох треків за допомогою *Sound Cue* і названо *AmbientMusicCue* (рис. 7).

Після цього було створено об'єкт класу *Actor*, де прописана логіка для старту та закінчення програвання музики. Також є обов'язковим додати користувачеві можливість вмикати та вимикати фонову музику на клавішу *M*.

Для атмосферного вступу у віртуальне середовище, створено пост-процесінговий матеріал, що містить блакитну сітку на чорному фоні, щоб із самого початку надати

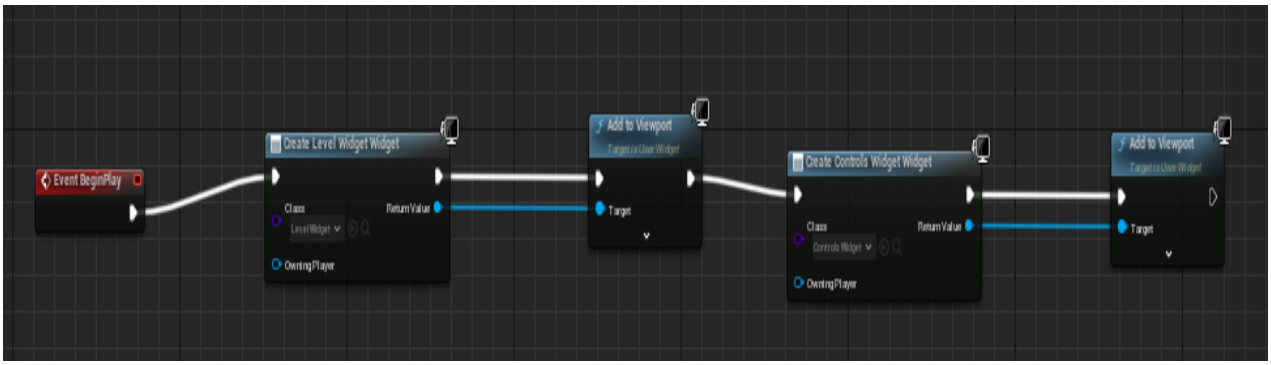


Рис. 5. Логіка виводу віджетів на екран

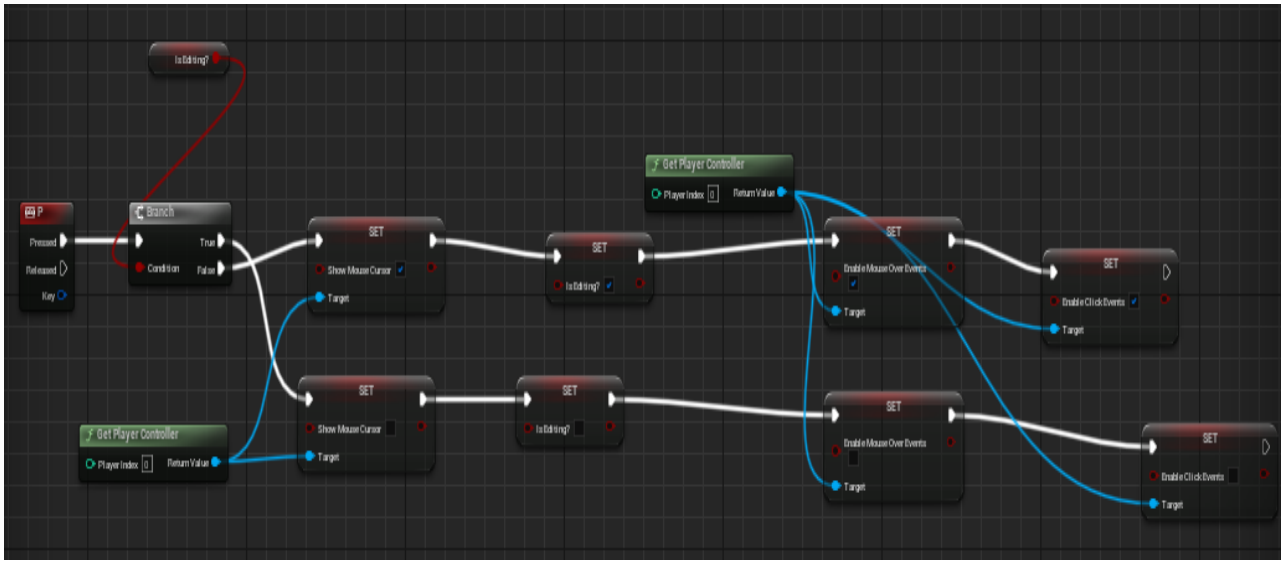


Рис. 6. Логіка вмикання та вимикання режиму редагування

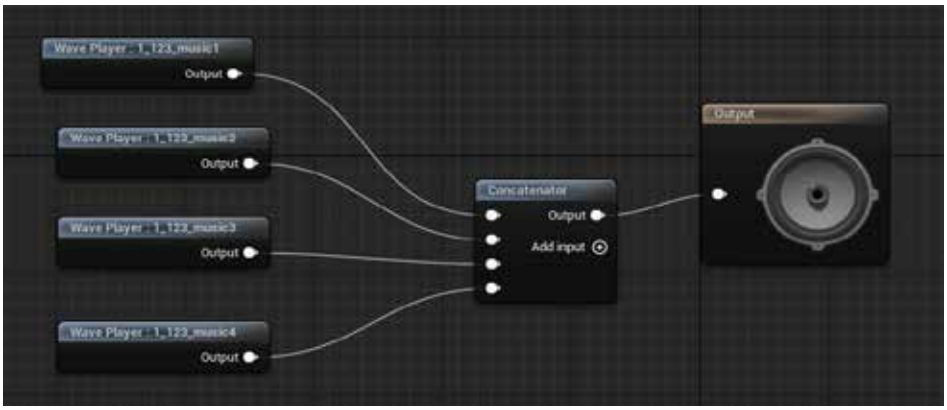


Рис. 7. Музикальна черга *AmbientMusicCue*

користувачеві розуміння того, що перед ним – саме віртуальне середовище, а не фотографія чи прирендер. Також було сформовано набір параметрів для цього матеріалу та створено параметр *Radius*. Після створення матеріалу та набору параметрів, було сформовано його *Material Instance* та додано до ще одного заздалегідь створеного *PostProcessVolume* через розділ *Post Process Materials*. А щоб фонова музика запрацювала з самого початку проекту та щоб відбувся ефект хвилі із сітки, створено логіку, що під'єднана до події *Event BeginPlay* у *BP_FirstPersonCharacter* ( ).

Щоб надати проекту ще більшого реалізму, було вирішено сформувати ефект зйомки з нагрудної камери. Для цього створено два об'єкти класу *CameraShake* і названо *Idle* та *Walk*. Ці об'єкти симулюють трясіння камери при статичному положенні та при ходьбі. Також, було створено пост-процесінговий матеріал, що створює ефект опуклої лінзи і ідентифіковано *M_Fisheye*, після чого створено його *Material Instance* та додано до заздалегідь доданого *PostProcessVolume* під назвою *PostProcessCamera* у *BP_FirstPersonCharacter*. Так як ефект зйомки з відеокамери повинен відбуватись постійно, було сформовано подію, яку далі була прив'язана до події *Event Tick*. А до кнопки що вмикає/вимикає цей ефект було прив'язано лише з задання стану симуляції (змінна *Videorecording Simulation*) та зміна стану застосування *PostProcessCamera*.

Демонстрацію візуального функціоналу сформованого *ArchViz* додатку через його скріншоти представлено на рис. 8. Розроблена модель має великий потенціал для застосування у представленні віртуальних моделей кінцевого продукту різним споживачам. Пропонуючи інтерактивне дослідження простору віртуальної аудиторії, користувачі мають широкі можливості взаємодіяти з різними елементами створеного середовища та маніпулювати ними,

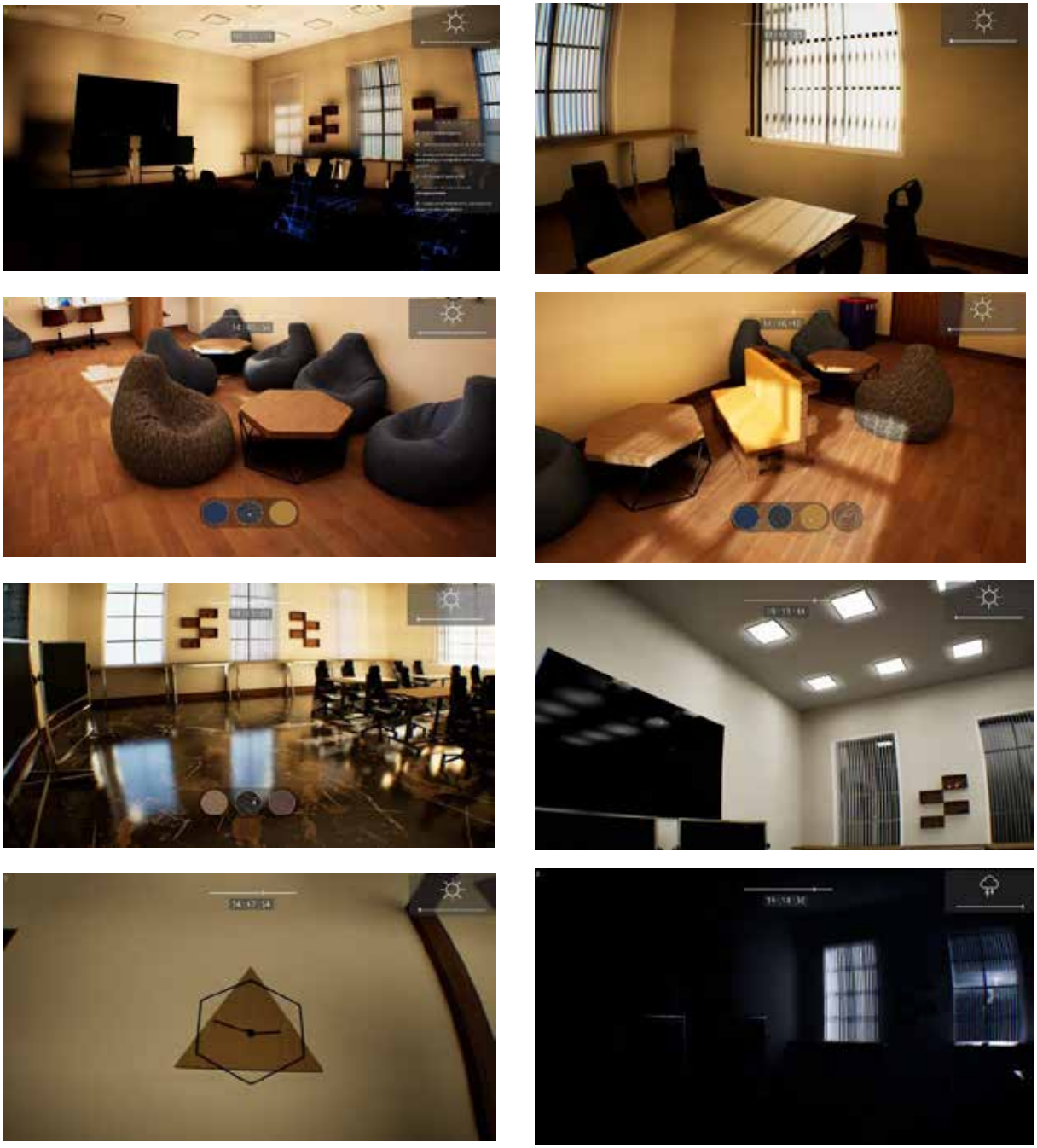


Рис. 8. Скріншоти *ArchViz* додатку з демонстрацією функціоналу

отримуючи глибше розуміння динамічних властивостей сформованого простору та його потенційного використання.

Висновки. Проведені експериментальні дослідження підтверджують позитивний прогноз подальшого прогресу у сфері розробки та використанні віртуальних моделей на базі додатків *ArchViz*. Таким чином, формується встановлення нових вимог до якості віртуального моделювання на основі використання досягнень в області технологій комп'ютерної графіки та інтерактивності, що надаються *Unreal Engine 5*. Розсуваючи межі можливого, очікується, що майбутні розробки ще більше підвищать реалістичність, занурення та зручність використання додатків *ArchViz*, встановлюючи нові стандарти в цьому напрямі для галузі інформаційних технологій.

ЛІТЕРАТУРА:

1. Ultra Dynamic Sky – Product Video + Quick Start (UE5 Version): https://youtu.be/b52npy-XUdQ
2. Навчальний посібник з Unreal Engine 5 для початківців – початковий курс UE5: https://youtu.be/k-zMkzmduqI
3. Unreal Engine 5 Import A Scene By Datasmith From 3DS Max 2022.2: https://youtu.be/kSDDetL3bYg
4. Master Material Basics! Unreal 5 (Must KNOW workflow!): https://youtu.be/tT0ANO5sXso
5. Material Parameter Collection | 5-Minute Materials [UE5]: https://youtu.be/J2Qf5v9_uSY
6. How to convert placed actors/meshes into blueprints UE5: https://youtu.be/MSYki36PJh8

REFERENCES:

1. Ultra Dynamic Sky – Product Video + Quick Start (UE5 Version): https://youtu.be/b52npy-XUdQ
2. Unreal Engine 5 Tutorial for Beginners – UE5 Starter Course: https://youtu.be/k-zMkzmduqI
3. Unreal Engine 5 Import A Scene By Datasmith From 3DS Max 2022.2: https://youtu.be/kSDDetL3bYg
4. Master Material Basics! Unreal 5 (Must KNOW workflow!): https://youtu.be/tT0ANO5sXso
5. Material Parameter Collection | 5-Minute Materials [UE5]: https://youtu.be/J2Qf5v9_uSY
6. How to convert placed actors/meshes into blueprints UE5: https://youtu.be/MSYki36PJh8