Національний університет «Полтавська політехніка імені Юрія Кондратюка»

(повне найменування вищого навчального закладу)

\_Навчально науковий інститут інформаційних технологій та робототехніки\_

(повна назва факультету)

\_Кафедра комп’ютерних та інформаційних технологій і систем\_

(повна назва кафедри)

**Пояснювальна записка**

**до дипломного проекту (роботи)**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_бакалавра\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(освітньо-кваліфікаційний рівень)

на тему

«Розроблення апаратно-програмних засобів віртуального музею «Поле \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Полтавської битви»»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Виконав: студент 4 курсу, групи \_401-ТК\_

спеціальності

\_\_123 Комп’ютерна інженерія \_

(шифр і назва напряму)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Сальніков С.В.\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

Керівник \_\_\_\_\_Беседін В.Ф.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

Рецензент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

Полтава – 2021 року

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**«ПОЛТАВСЬКА ПОЛІТЕХНІКА ІМЕНІ ЮРІЯ КОНДРАТЮКА»**

**НАВЧАЛЬНО НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА РОБОТОТЕХНІКИ**

**КАФЕДРА КОМП’ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І СИСТЕМ**

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**

**спеціальність 123 «Комп’ютерна інженерія»**

**на тему**

**«Розроблення апаратно-програмних засобів віртуального музею «Поле Полтавської битви»»**

**Студента групи 401-ТК Сальнікова Святослава Віталійовича**

Керівник роботи

доктор економічних наук,

професор Беседін В.Ф.

Консультант

кандидат технічних наук,

доцент Дмитренко Т.А.

Завідувач кафедри

Полтава – 2021

**РЕФЕРАТ**

Кваліфікаційна робота бакалавра: 56 с., 11 малюнків, 1 додаток, 14 джерел.

**Об’єкт дослідження**: апаратно-програмні засоби віртуального музею «Поле Полтавської битви

**Мета роботи**: розроблення віртуального музею «Поле Полтавської битви», а саме створення тримірної моделі з можливістю вільного переміщення та всебічного огляду будівлі, що дозволить відвідувати музейний заклад у дистанційній формі.

**Методи**: проектування та розробка тримірної моделі музею, використання програмних засобів ігрового рушія Unity для переміщення між залами.

**Ключові слова**: модель, експозиція, текстура, музей, скрипт, сцена, проект.

# 

**ЗМІСТ**

ПЕРЕЛІК УМОВНИХПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ …………...4

ВСТУП ….5

РОЗДІЛ 1 АНАЛІТИЧНИЙ ОГЛЯД СИСТЕМИ ВІРТУАЛЬНОГО МУЗЕЮ 8

1.1 Опис предметної області ………………………………………….. 8

1.2 Огляд джерел та існуючих рішень……………………………………. 14

1.3 Специфікація вимог………………………………………… 21

РОЗДІЛ 2 ПРОЕКТУВАННЯ АПАРАТНО-ПРОГРАМНИХ ЗАСОБІВ ВІРТУАЛЬНОГО МУЗЕЮ 23

2.1 Загальна постановка і класифікація задач сучасного проектування .23

2.2 Проектування моделі музею …………………………………………...25

2.3 Проектування залів музею ……………………………………………..29

РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОГО ЗАСТОСУНКУ ВІРТУАЛЬНОГО МУЗЕЮ 34

3.1 Вибір платформи ………………………………………………………34

3.2 Постановка сцени ……………………….……………………………..37

3.3 Робота з камерою……………………………………………………….39

ВИСНОВКИ 46

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 48

ДОДАТОК А ТЕКСТ СЦЕНАРІЮ ПОВЕДІНКИ КАМЕРИ 50

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

* **Мультимедіа** – комбінування різних форм подання інформації за допомого одного носія, наприклад, текстової, звукової, графічної, анімації та відео; комп’ютерна технологія, яка поєднує різні типи медіа, такі як текст, аудіо, відео, анімація.
* **QR-код —** матричний код (двовимірний штрих-код), основною перевагою якого є легке розпізнавання сканувальним обладнанням (в тому числі й фотокамерою мобільного телефона)
* **ІТ** – інформаційні технології
* **МР3** – формат файлу для зберігання аудіоінформації.
* **C#** – читається “сі-шарп”; об’єктно-орієнтована мова програмування високого рівня
* **Рéндеринг** — в комп'ютерній графіці — це процес отримання зображення за моделлю з допомогою комп'ютерної програми
* **FBX** - це власний формат файлу (.fbx); використовується для забезпечення взаємодії між програмами для створення цифрового вмісту
* **Скрипт** – сценарій поведінки об’єкту
* **IDE** – Інтегроване середовище розробки; комплексне програмне рішення для розробки програмного забезпечення

**ВСТУП**

**Актуальність роботи.** У сучасному суспільстві туризм відіграє далеко не останню роль у культурному розвитку, в той час як музеї є невід’ємною його частиною. Загальні світові тенденції заохочують людей до подорожей і відвідування туристичних місць, проте у зв’язку з останніми подіями дістатися до бажаної точки стало значно складніше. Великої уваги в такий період слід приділити розробці нових та покращенню вже існуючих способів донесення до людей культурної значущості тих чи інших музеїв до найбільш зацікавленої категорії громадян, тобто молоді. Значна міра віртуалізації сучасного суспільства значною мірою сприяє розвитку таких методів, адже з інформаційним розвитком суспільства зростає кількість людей, що надають перевагу мережі Інтернет порівняно з іншими джерелами інформації [1].

Сучасні музеї використовують різні методи для наближення до сучасної молоді та розширення власних можливостей. Серед них є два основних напрями:

Перший напрям – установка біля туристичних об’єктів та експонатів цифрових засобів передачі інформації.

Другий напрям – створення апаратно-програмних засобів поширення інформації мережею інтернет. Даний метод є дуже важливим і актуальним, адже спрямований на спрощення відвідування музейних закладів та отримання актуальної інформації щодо них, а також на їх популяризацію серед широкої аудиторії людей різних категорій.

**Основною метою** розроблення віртуального музею є підвищення спроможності людей оглядати заклад дистанційно та підвищення його популярності серед інтернет-користувачів. Цієї мети можна досягти завдяки сучасним апаратним та програмним засобам, що дозволяють швидко та якісно створювати об’ємні моделі та текстури необхідної якості і ступеню деталізації, а також програмне забезпечення, яке дозволяє переміщуватися по створеній моделі.

**Тема** дипломної роботи присвячена розробці апаратно-програмних засобів віртуального музею «Поле Полтавської битви». На сьогодні віртуальні музеї є доволі популярним рішенням. Такі системи дозволяють відвідувати музейні заклади дистанційно, що популяризує культурний розвиток та призводить до збільшення потоку туристів та покращення трудового процесу працівник галузі культури.

Власне такі проекти допомагають людям з усіх кутків нашої планети отримати інформацію про те чи інше культурне надбання людства. Не всі люди мають можливість вільно подорожувати і вивчати культуру та побут інших народів, що пов’язано з браком коштів та часу. Проте навіть ті люди, які можуть собі дозволити такі подорожі, не в змозі осягнути всієї інформації, що міститься в музеях хоча б однієї країни чи навіть одного міста. Наприклад, музейні фонди України містять понад 12 млн. експонатів. На перегляд в музеях виставляється лише невелика кількість експонатів, а інша, набагато більша частина зберігається в музейних сховищах. Така ситуація зумовлена, по-перше, нестачею місця для проведення нових виставок в приміщеннях музею чи інших пристосованих для цього приміщеннях, а, по-друге, умовами зберігання експонатів, оскільки багато з них дуже чутливі до різних природних чинників, а недотримання умов зберігання може спричинити їх повну або часткову руйнацію.

Враховуючи складність таких систем, час підготовки всього необхідного та вимоги до обладнання, зазвичай їх створення займає великі обсяги часу та сил. У зв’язку з цим та відсутністю професійного обладнання, було прийнято рішення спростити процес розробки та створення моделі, відмовившись від вільного переміщення залами та зменшивши якість текстур. Таке рішення дозволяє скоротити час розробки та полегшити процес роботи, проте дещо зменшити якість кінцевого продукту на даному етапі.

Враховуючи вищезазначене, варто відмітити, що даний програмний засіб може бути використаний для просування культурного розвитку, популяризації туризму та розширення кола потенційних відвідувачів музею Полтавської битви у майбутньому.

**РОЗДІЛ 1**

**АНАЛІТИЧНИЙ ОГЛЯД СИСТЕМИ ВІРТУАЛЬНОГО МУЗЕЮ**

**1.1 Опис предметної області**

Великою популярністю користується національна ідея та пізнання справжньої історії країни серед людей різних категорій. Проте залишаються проблеми, що часто стають на заваді розширенню кругозору людей шляхом відвідування музеїв. Серед таких проблем можна зазначити відстань (музеї розкидані по всій країні), мовний бар’єр (у випадку з іноземними гостями), обмежений особистий бюджет (не кожен може дозволити собі послуги гіда) тощо. Це у свою чергу стимулює розвиток туристичної галузі та створення все нових і відновлення існуючих методів донесення інформації до суспільства, минаючи перераховані проблеми.

Так музеї, котрі зазвичай і є головними туристичними точками, оновлюють свої експозиції, привносять в них нові та сучасні рішення, а також використовують плоди технологічного прогресу для спрощення донесення інформації до відвідувачів. Більшість з цих нововведень дає змогу отримати повний обсяг необхідної інформації, навіть не користуючись послугами гіда. Серед них можна виділити наступні варіанти:

* аудіогід – аудіоплеєр з приєднаними до нього навушниками та аудіозаписом тексту екскурсії, зазвичай декількома мовами);
* інформаційні дошки – таблички з певною кількістю інформації, розміщені поруч з експонатами;
* QR-коди – схожі до інформаційних дошок, але на таких табличках розміщено лише QR-код, зчитування якого за допомогою мобільного додатку на смартфоні відкриє вам сторінку в інтернеті, де буде знаходитися інформація про експонат або туристичний об’єкт;
* туристичні кіоски – зазвичай планшети або термінали різного розміру (від звичайного з діагоналлю 7 дюймів, до великих пристроїв, що нагадують телевізори), скориставшись якими можна отримати доступ до великого обсягу інформації, яка відрізняється залежно від місця розміщення кіоску та його призначення;
* віртуальні музеї – повністю відзняті моделі музеїв з можливістю оглядати різні зали та експонати у вигляді псевдо-об’ємних панорам, що повністю відтворюють інтер’єр будівлі, створюючи ефект присутності.

Кожен з цих методів існує вже досить довгий час і користується популярністю в ряді музеїв та туристичних об’єктів, зокрема і в Україні. Кожен з них дає змогу вирішити певні проблеми доступності до туристичних даних в тій чи іншій мірі. Аудіогіди, інформаційні дошки та QR-коди пройшли випробування часом і показали себе як надійні способи наблизити різні категорії людей до пам’яток культури. Всі ці три варіанти підвищення доступності до культурного розвитку знайшли своє місце у багатьох містах і зараз їх можна знайти у більшості музеїв та біля багатьох важливих історичних пам’яток, тобто вони стали невід’ємною частиною туристичної частини міст, чим помітно розширили можливості культурного збагачення не лише гостей, а й жителів міст.

Туристичні кіоски також є важливою складовою для поширення культурного розвитку, адже з їх допомогою будь-який турист може отримати інформацію про всі музеї та важливі пам’ятки міста та області, готелі, місця, де можна придбати сувенірну продукцію та багато іншого. Такі термінали зазвичай встановлюють біля популярних туристичних об’єктів або на території музейного комплексу. Також існує варіант зі встановленням більш компактного варіанту у вигляді планшету, на екрані якого завжди відображається певна інформація про музей загалом, певний зал, експонати, історичні особистості тощо. Не буде помилкою відносити їх до різновиду інформаційних дошок.

Найкращим же варіантом вирішення проблеми доступності культурного збагачення слід вважати віртуальні музеї, котрі поєднують у собі можливості попередніх перерахованих засобів. Таким чином, якісно створений віртуальний музей може повністю замінити реальне відвідування музею, при цьому не вимагаючи від відвідувача подорожувати чи витрачати свої кошти (зазвичай, перегляд віртуальних музеїв безкоштовний і повністю вільний). Варто зазначити, що не всі музеї створюють свою цифрові аналоги користуючись однаковими методами.

У мережі Інтернет віртуальний музей визначається як колекція цифрових зображень, звукових файлів, текстових документів та інших даних історичної, наукової чи культурної цінності, які є доступними через електронні медіа. У віртуальному музеї можна побачити значну кількість експонатів, які з різних причин не можуть бути виставлені на показ у реальному музеї.

З поняттям віртуального музею тісно пов’язане поняття віртуальної галереї. На жаль, офіційного визначення цього терміну немає, тому віртуальною галереєю вважатимемо частину віртуального музею, в якій зберігається та відображається вся мультимедійна інформація.

На основі вищеописаного віртуальна галерея володіє функціями перегляду і/або прослуховування мультимедійних даних (фото, аудіо, відео, анімація, 3D-зображення та ін.). За допомогою фото-, відео- та аудіоматеріалів віртуальної галереї можна переглянути експонати, прослухати їхню історію та оцінити їхній стан, тобто отримати вичерпну інформацію про експонати. Наприклад, якщо під експонатом розуміється сільськогосподарське чи інше знаряддя, то відеосюжет може містити інформацію про період часу та місце його побутування; ким, з якою метою та яким чином він використовувався; де і коли був знайдений. Крім того, додатковою функцією віртуальної галереї є перегляд відеоматеріалів про експонати музею із можливістю сурдоперекладу, яка, своєю чергою, дасть можливість людям із вадами слуху і/або мови дізнатися більше про музейні експонати. Аудіозаписи ж можуть відображати звуки деяких експонатів музею. Наприклад, такими звуками можуть бути мелодія музичної шкатулки, спів птахів і т.ін. Крім того, для підвищення свого рейтингу та авторитету віртуальна галерея має функцію віртуальної екскурсії, яку проводить віртуальний екскурсовод.

Говорячи про розробку віртуального музею «Поле Полтавської битви», необхідно згадати також історію самого музею, аби краще розуміти, з чим ми маємо справу. Історія музею розпочалася в дні помпезного святкування 200-річчя Полтавської битви. Поразка в російсько-японській війні, революція 1905-1907 рр., Відродження національних рухів спонукали владу посилити імперсько-патріотичне виховання населення шляхом пропаганди військово-історичного минулого, прославлення могутності Російської держави та її армії. Однією з найвигідніших тем для цього стала перемога Петра I у 1709 р. Імперія продовжувала створювати міф про битву під Полтавою як "славну Вікторію". У серпні 1908 року Микола II підписав указ про створення Міжвідомчої комісії з підготовки святкування 200-річчя Полтавської битви на чолі з генералом кавалерії бароном А. Більдерлінгом. Місцем проведення низки запланованих урочистостей була Полтава. До ювілейної дати в місті встановили пам'ятники, а біля Шведської могили відкрили музей Полтавської битви [2] .

Засновником та першим директором (1909-1918) музею був відомий полтавець Іван Павловський (1851-1922). Тридцять дев’ять років викладав історію в Петровському Полтавському кадетському корпусі (1874-1913). Вчений був одним із засновників і постійним секретарем Полтавської губернської наукової архівної комісії (1903-1918), активним членом Полтавської церковної історико-археологічної комісії (1906-1917), неодноразово обирався голосним Полтавської міської думи . Сучасні історики досі користуються бібліографічними довідниками з історії Полтавщини, виданими І. Павловським.

У каталозі Музею Полтавської битви на шведській могилі, виданому в 1910 році, видно, що колекція складалася з 339 експонатів: портретів історичних діячів, гравюр, картин, зброї та манекенів солдатів. Притаманне І. Павловському прагнення повністю висвітлити події сприяло наповненню експозиції предметами, відображало всі сторони військового конфлікту. Тому відвідувач міг також побачити портрети шведських воєначальників, козацькі старожитності та дев’ять зображень гетьмана Івана Мазепи (за радянських часів на виставці не було жодного).

Протягом 1917-1918 років музей зазнав декількох пограбувань, а І. Павловський передав залишки колекції до Центрального пролетарського музею Полтави (нині Полтавський краєзнавчий музей імені Василя Кричевського). У 20-30-ті роки він виставлявся в окремій залі під назвою "Полтавська битва" і протягом двох десятиліть зазнавав діаметрально протилежних оцінок. Таким чином, в період українізації висловлювалася думка ліквідувати цю частину музею, "тому що ... для України цей момент є особливо трагічним, оскільки з тих пір царська Москва міцно стиснула Україну під себе і позбавила споконвічної автономності вільнолюбний український народ". І вже в 30-х роках радянські ідеологи почали активно впроваджувати тезу про братнє героїчне минуле російського, українського та білоруського народів, і тоді особливої актуальності набувають Події 1709 р., «Коли фашистські воєначальники мріють захопити процвітаючу соціалістичну Україну”. На початку 1939 р. До 230-ї річниці Полтавської битви були видані загальносоюзні та республіканські постанови КП (б), КП (б) У. Музейна виставка "Полтавська битва" значно розширена.

Під час Другої світової війни у вересні 1941 року частина колекції І. Павловського була евакуйована до Уфи, решта предметів залишилася в Полтаві та постраждала від пожежі під час нацистської окупації. І вже в липні 1944 року в краєзнавчому музеї полтавці могли ознайомитись з виставкою «Героїчне минуле нашого народу», в якій був розділ «Полтавська битва».

У травні-червні 1949 року постановами Рад Міністрів СРСР та Української РСР він зобов'язаний "відновити Музей Полтавської битви, як музей республіканського підпорядкування другої категорії". Раптово заклад відкрили 23 вересня 1950 року в приміщенні інвалідного будинку на Шведській могилі. Сюди повернулися вцілілі предмети з колекції І. Павловського. Колекція також поповнилася унікальними артефактами початку 18 століття завдяки надходженням від інших музейних установ колишнього Радянського Союзу, зокрема від Ермітажу, Третьяковської галереї, Київського, Львівського, Чернігівського історичних музеїв.

У повоєнний період за наказом Сталіна вийшла монографія провідних радянських істориків Б. Тельпуховського, Є. Порфир'єва, В. Шутого, Е. Тарле, присвячена темі "захисту" Росії від Карла XII. У радянському міфі про події Північної війни в Україні з'явилися поняття "народна війна", "партизанська війна проти шведських загарбників", які поєднувалися з "класовою боротьбою народу проти експлуататорів-мазепинців". Збільшився наголос на збереженні лояльності українського народу до союзу з Росією. Ці обставини суттєво вплинули на тематично-експозиційний план музею. З нагоди 250-ї річниці історичної події тут було добудовано додатковий зал, в якому діорама "Полтавська битва", виконана художниками військової майстерні М. Грекова, та бойові полотна майстрів радянської епохи, демонструючи героїчний міф про Полтавську Вікторію, були розміщені. За радянських часів музею відводилася роль активного пропагандиста комуністичної ідеології.

У другій половині 80-х років ХХ століття під керівництвом заслуженого художника України А. Щербака музей було повторно експоновано. Збережений хронологічний принцип викладу подій Північної війни та висвітлено вплив Полтавської битви на хід європейської історії. Відзначено участь України у військовому протистоянні. У 1994 році в музеї було відкрито зал, присвячений історії українського козацтва. Завдяки співпраці з Товариством військової історії Швеції, експозиція поповнилася предметами, які розкривають участь Калолінів у подіях Північної війни. Враховуючи опубліковані історичні джерела та дослідження провідних українських істориків, екскурсійне забезпечення музею постійно поповнюється новими матеріалами.

Зараз колекція налічує понад сім тисяч експонатів, з них три тисячі двісті - предмети основного фонду. Експозиція містить унікальні картини Дениса Мартіна (молодшого), Бернардо Белотто Каналетто, Жана Марка Ната, Юрія Рєпіна, Миколи Самокіша, оригінальні портрети історичних діячів художників 17 століття, гравюри, автентичну зброю, археологічні знахідки, старовинні предмети побуту.

Отже, туристична сфера вже активно використовує розробки ІТ для популяризації культурного розвитку та підвищення її доступності для різних категорій громадян. Тому слід продовжити розробку та покращення технологічних надбань у даній галузі для подальшого їх впровадження у культурний процес.

**1.2 Огляд джерел та існуючих рішень**

Розвиток сучасних віртуальних музеїв неможливий без урахування попередніх розробок та досвіду, накопиченого в даній галузі. Тому дослідження стану питання віртуальних музеїв є актуальним і необхідним завданням у межах проекту «Розроблення апаратно-програмних засобів віртуального музею «Поле Полтавської битви»».

Згідно з власним дослідженням автора роботи, можна виділити наступні основні види віртуальних музеїв:

* онлайн-картотека – фактично, онлайн-бібліотека, що містить так звані паспорти експонатів та реліквій, представлених на експозиції музею чи музейного комплексу, та/або тих, що містяться в його фондах. Такі форми віртуальних музеїв з’явилися досить давно і набули певної популярності за час свого існування;
* псевдомодель, створена за допомогою широкоформатних панорамних фото. Більшість віртуальних турів до історичних місць створені саме таким способом. Пересування таким музеєм відбувається між заданими точками, кожна з яких є центром панорамної сфери, котру відвідувач може повертати, але не може переміщуватися по ній;
* віртуальна тримірна модель музею з використанням фотографічних текстур, створена за допомогою програмних засобів і яка своєю структурою нагадує комп’ютерну гру. Даний вид віртуального музею є початковою ідеєю даного дипломного проекту, проте через складність виконання довелося спростити деякі задумані механіки, а від інших відмовитися.

Всі перераховані види віртуальних музеїв нині існують і активно використовуються у сфері туризму для заохочення людей до культурного збагачення та надання вільного доступу до актуальної інформації щодо роботи музеїв та їх експозицій. У переважній більшості такі музеї суміщають у собі технології 3D-моделювання, цифрової обробки фотографій, мережевого програмування HTML, CSS та ін. Говорячи про види віртуальних музеїв, слід привести приклади їх використання.

Перш за все, онлайн-картотеки. Це зручний спосіб роздобути цікаву і корисну інформацію, необхідну для роботи чи навчання, не виходячи з дому. У якості яскравого прикладу такої форми віртуального музею можна привести Токійський національний музей (Японія) [3]. Його сайт одразу зустрічає гостя яскравою та зрозумілою головною сторінкою, що містить основну важливу інформацію для відвідувача і доступна одразу кількома мовами: активні виставки, новини музею, грядущі та діючі події, розклад роботи та інформацію про те, як зв’язатися з музеєм, дістатися до нього та замовити екскурсію. Також тут можна знайти список категорій, серед яких є наступні:

* Виставки – список всіх доступних виставок музею, їх короткий опис, час роботи, доступність та чому їх варто відвідати;
* Події – вся необхідна інформація про події, лекції, концерти, симпозіуми, волонтерські програми, воркшопи і т.д., що проходять чи плануються на території музею;
* Колекції – найцікавіша для нас категорія, саме в ній знаходиться інформація про експозицію та експонати музею, тому розглянемо її далі;
* Навчання – містить інформацію про всі навчальні програми, які представляє музей: шкільні, публічні, сімейні, учительські волонтерські і т.д.;
* Дослідження – інформація про роботу музею по збереженню та відновленню пам’яток історії, наукові дослідження, публікації та симпозіуми;
* Доступ – найважливіша категорія для всіх, хто збирається відвідати музей; інформація про години роботи, як дістатися до місця (громадським транспортом чи автомобілем), ціна відвідування та додаткових послуг (тут же можна замовити квитки), вихідні дні, способи зв’язатися з представниками музею.

Розглянувши перераховані категорії, стає складно назвати даний віртуальний музей простим, проте він насправді виконаний дуже просто і зрозуміло. Так і має бути оформлений хороший сайт подібного типу у такого музею – мінімалістично і максимально зрозуміло для людей всіх категорій та віку, адже в наш час досі багато хто досить погано розбирається в сучасних технологіях, тому надто яскравий і заплутаний, хоч і красивий сайт виявиться незручним і це вплине на їх оцінку роботи музею та бажання відвідати його, чого робітники культури ніяк не можуть допускати.

Найважливішою для нашого огляду категорією є «Колекції», зібрання музейних реліквій, розділених по підкатегоріям. Одразу ж в очі кидається пункт «e-Museum» («електронний музей») – це, фактично, цифрова експозиція, доступна до вільного перегляду з будь-якої точки світу у будь-який момент часу. Експонати (або, краще кажучи, реліквії), можна переглянути за наступними пунктами:

* Архітектура;
* Скульптура (різноманітні статуетки, скульптури, ляльки, маски театру «Но» тощо);
* Картини (давній живопис, гравюри, портрети та багато іншого);
* Каліграфія (древні сувої, що представляють собою пам’ятки давнього японського мистецтва художнього письма);
* Вироби з металу (різноманітні вироби з різних металів, часто ритуальні);
* Мечі (колекція древніх мечів, в основному самурайських, роботи різних майстрів, як іменитих, так і безіменних);
* Історичні Матеріали (документи та карти);
* Скарби Хорюджі (різні предмети, що мають релігійне значення);
* Археологія (викопні пам’ятки історії та культури);
* Текстиль (вироби із тканини, в більшості своїй одяг);
* Кераміка;
* Лаковані Вироби (вироби з дерева, шкіри чи інших матеріалів, вкритих лаком і які вважаються витворами мистецтва).

Кожен пункт містить від декількох до кількох десятків реліквій, для кожної з яких представлені фото високої якості, опис розмірів, ваги, періоду виготовлення, місця, де даний експонат було знайдено чи від кого отримано, а також невеликий опис його історії та призначення. До того ж варто відмітити, що більшість реліквій Токійського національного музею є національними скарбами або важливою культурною спадщиною. До всього цього також варто додати те, що у музею є віртуальна бібліотека, в якій можна знайти відскановані історичні документи та праці, якими він володіє.

Розглянувши приклад онлайн-картотеки, переходимо до псевдомоделі за допомогою широкоформатних панорамних фото. Більшість віртуальних турів до історичних місць створені саме таким способом. Створення віртуального туру саме в такій формі потребує відносно великих затрат часу та дуже якісне обладнання для зйомки та обробки панорам, аби вони точно передавали як насправді виглядає місцевість, зображена на них, та створювали відчуття присутності для глядача. В цілому, панорамна псевдомодель дозволяє наблизитися до пам’яток історії та культури, не виходячи з дому, побачити на власні очі експозицію музею та пройтися його залами. Також, до такого туру можна додати голосовий супровід фахівця, що створить відчуття екскурсії і дозволить якомога краще передати відчуття особистого відвідування. Проте у даного виду віртуалізації музею є й свої недоліки. Наприклад, якою б точною і широкоформатною не була панорама, ви не зможете вільно пересуватися нею, адже ви обмежені її центром – місцем зйомки, тобто у кожній точці наче знаходитесь усередині сфери; це не шкодить загальному сприйняттю, проте може завадити зблизька розглянути певні експонати.

Загалом, існує багато музеїв, які мають подібні віртуальні тури своєю територією, але хотілося б розглянути найбільш помітний серед них, а саме музейний комплекс Ермітаж [4]. Відвідуючи офіційний сайт цього неймовірного музею, ми можемо бачити не менш зручну і зрозумілу головну сторінку, ніж у Токійського національного музею, яка зустрічає нас основною важливою інформацією. Тут ми можемо побачити інформацію для відвідувачів, новини музею, анонси грядущих та інформацію про діючі виставки, архів з колекцією експонатів, наукові роботи та, найголовніше, віртуальний візит до цього велетенського музейного комплексу. У розділі «Колекції» знаходиться зібрання усього, що представлено на експозиції Ермітажу і все це представлено у вигляді паспортів, схожих за своєю структурою до аналогічних у Токійського національного музею, тому даний пункт ми не будемо розглядати детально. Нас цікавить саме віртуальний візит, тому перейдемо до нього.

Ермітаж представляє своєму віртуальному гостю доступ до всіх своїх музейних комплексів, а саме головної будівлі, головного штабу, зимового палацу Петра I, палацу Меншикова, Ермітажного театру, музею імператорського фарфорового заводу та реставраційно-депозитарного комплексу. Окрім цього, нам доступні галерея дорогоцінностей, різноманітні виставкові проекти, зовнішній вигляд всього комплексу ззовні та виставкові центри. Загалом, віртуальна модель Ермітажу складається з понад однієї тисячі широкоформатних панорам, кожна з яких є одним повним або частиною великого залу та дозволяє оглянути їх з усіх боків (можна навіть розглянути стелю та підлогу, тобто глядач насправді наче знаходиться у сфері), а їх кількість та спосіб поєднання в єдине ціле виконані таким чином, що ви не пропустите жодного експонату, жодної реліквії та жодної деталі. Заходячи до зали ми бачимо мітки на важливих експонатах або просто на стінах. Натиснувши на них, можна дізнатися, що саме було в тій чи іншій кімнаті за часів, коли музей ще був палацом, які історичні події з ним пов’язані. Те ж саме стосується і експонатів – натискання на їх мітки дозволить зблизька розглянути обрану реліквію і дізнатися про неї щось цікаве.

Іншим не менш цікавим представником панорамного віртуального музею є Смітсонівський національний музей природничої історії, який представляє відвідувачу можливість пройтися його залами, використовуючи широкоформатні панорамні фото [5]. Загальна структура майже не відрізняється від Ермітажу за винятком того, що мітки на певних експонатах дозволяють просто наблизити зображення, але не надають якоїсь інформації про нього (проте це й не потрібно, вона зазвичай присутня на самому фото).

Серед українських же музеїв варто згадати такий проект як «Музеї України просто неба» [6]. Він зібрав у собі 7 музейних комплексів, кожен з яких представлений серією панорам зі спокійним музичним супроводом та розповіддю голосом професійного диктора. У рамках проекту відвідувачу надається можливість відвідати наступні комплекси:

* Музей народної архітектури та побуту (м. Ужгород);
* Музей народної архітектури та побуту «Шевченківський гай» (м. Львів);
* Музей народної архітектури та побуту Середньої Наддніпрянщини (м. Переяслав-Хмельницький);
* Мамаєва Слобода (м. Київ);
* Національний музей народної архітектури та побуту України (м. Київ);
* Резиденція Богдана Хмельницького (м. Чигирин);
* Запорізька Січ (м. Запоріжжя).

Варто зазначити, що саме таке поєднання якісної віртуальної моделі та розповіді про місце відвідування надає найкраще відчуття подорожі до того чи іншого музею, наче ви побували там насправді, пройшлися коридорами та залами будівлі чи доріжками території та почули історії місця від професійного знавця.

Ми розглянули основні види віртуальних музеїв і тепер настав час перейти до того, що цікавить нас найбільше – віртуальна тримірна модель музею з використанням фотографічних текстур. Саме в такому вигляді планується створення віртуального музею «Поле Полтавської битви». Такий спосіб віртуалізації не є розповсюдженим, але й назвати його чимось абсолютно новим не вийде. Так, знайти конкретний приклад складно, але можливо, і це синагога у місті Жовква [7]. Дане рішення є дещо відмінним від задуманого у рамках дипломного проекту, але дуже близьке за змістом. Говорячи про синагогу, потрібно сказати, що це дуже якісна віртуальна модель, що повністю повторює розміри та форму справжньої будівлі, а фотографічні текстури дозволяють побачити, як саме вона виглядає на даний момент. Проте екскурсія цією будівлею виконана у вигляді відео, в якому камера сама переміщується від однієї точки до іншої.

Отже, як можна зрозуміти з усього вищесказаного, усі методи віртуалізації музеїв мають спільні риси, але сам процес та кінцевий результат помітно відрізняються. Це дозволяє вирішити, як саме потрібно організувати розробку проекту та які методи розробки буде доцільно використовувати у випадку створення віртуального музею «Поле Полтавської битви».

**1.3 Специфікація вимог**

Опираючись на всі перераховані вище методи створення віртуальних музейних турів, розглянемо, що саме потрібно для розробки нашого проекту віртуального музею «Поле Полтавської битви».

Для створення мінімального ефекту присутності, необхідно створити тримірну модель музею, зберігаючи реальні масштаби, а також фотографії достатньої якості, які будуть використані у якості текстур. Тобто, сама будівля музею використовується як локація, фотографічні текстури передають загальний вигляд залів та дозволяють оглядати загальний вид залів, а ігровий рушій дає відвідувачу змогу переміщуватися між залами за допомогою вільної камери та оглядати будівлю з усіх боків, тобто як інтер’єр, так і екстер’єр. Створення такого віртуального музею несе за собою значно менші затрати часу і потребує набагато менш потужного обладнання, ніж якби створювалася детальна модель з відтворенням усіх експонатів, представлених на експозиції.

Кінцевий продукт повинен мати вигляд простої гри, або, скоріше, застосунку, інтерактивної тривимірної моделі, у якій користувач матиме змогу побачити зали музею історії Полтавської битви, розглянути експонати, представлені на його експозиції. Задача даного проекту полягає у популяризації культурного розвитку та туризму серед сучасної молоді і не тільки, розширенні потенційної аудиторії музею історії Полтавської битви та підготовка їх до туристичного візиту до реального музею.

**РОЗДІЛ 2**

**ПРОЕКТУВАННЯ АПАРАТНО-ПРОГРАМНИХ ЗАСОБІВ ВІРТУАЛЬНОГО МУЗЕЮ**

**2.1 Загальна постановка і класифікація задач сучасного проектування**

Враховуючи специфіку проекту та вимоги, встановлені щодо його виконання, процес проектування та підготовки до використання може бути досить тривалим. Підготовка цифрових матеріалів займає певний час та вимагає певної підготовки, правильного вибору обладнання та моменту зйомки. Саме ж проектування здійснюється на основі даних та креслень, отриманих від співробітників музею історії Полтавської битви. Задля пришвидшення процесу та підвищення якості зовнішнього вигляду кінцевої програми, вирішено використовувати наступні програмні засоби:

* Adobe Photoshop – це редактор растрової графіки, розроблений та виданий Adobe Inc. для Windows та macOS. Спочатку він був створений в 1988 році Томасом та Джоном Ноллами. З тих пір програмне забезпечення стало галузевим стандартом не лише редагування растрової графіки, а й цифрового мистецтва в цілому. Таким чином, назва програмного забезпечення стала загальним товарним знаком, що призвело до використання його як дієслова (наприклад, "для фотошопу зображення", "фотошопінг" та "конкурс фотошопів"), хоча Adobe не рекомендує такого використання. Photoshop може редагувати та складати растрові зображення у декілька шарів та підтримує маски, альфа-композитування та декілька кольорових моделей, включаючи RGB, CMYK, CIELAB, кольорові плями та двотонний звук. Photoshop використовує власні формати файлів PSD та PSB для підтримки цих функцій. На додаток до растрової графіки, Photoshop має обмежені можливості редагування або відтворення тексту та векторної графіки (особливо через відсічний контур для останніх), а також 3D-графіки та відео. Його набір функцій можна розширити за допомогою плагінів (програми, розроблені та розповсюджені незалежно від Photoshop, що працюють у ньому та пропонують нові або вдосконалені функції) [8];
* SketchUp – програма для 3D дизайну та архітектурного проектування. В основному використовується для моделювання житлових будинків, меблів, інтер'єру. Є інструменти для проектування драбин, електропроводки, санітарно-технічних комунікацій та обладнання. Однак існують і значно масштабніші проекти на її основі. Так, наприклад, у SketchUp була створена 3D-модель міста Красноярська з геодатою. У березні 2006 року компанія SketchUp придбала компанію Google разом із невеликою фірмою @Last Software. У квітні 2012 року Google продав компанію SketchUp компанії Trimble Navigation. Існують дві версії програм – безкоштовна для некомерційного використання, обмежена за функціональністю SketchUp Make (перш за все у плані експортування в інші формати) та платна SketchUp Pro [9];
* Unity – це крос-платформний ігровий рушій, розроблений Unity Technologies, вперше оголошений і випущений у червні 2005 року на всесвітній конференції розробників Apple Inc. як ексклюзивний ігровий движок для MacOS X. З тих пір двигун поступово розширюється для підтримки різноманітних настільних, мобільних, консольних та віртуальних платформ. Він особливо популярний для розробки мобільних ігор для iOS та Android і використовується для таких ігор, як Pokémon Go, Monument Valley, Call of Duty: Mobile, Beat Saber та Cuphead. Він цитується як простий у використанні для початківців розробників і популярний для розробки ігор Indie. Рушій можна використовувати для створення тривимірних (3D) та двовимірних (2D) ігор, а також інтерактивних симуляцій тощо. Рушій був прийнятий на виробництво за межами відеоігор, таких як кіно, автомобільна промисловість, архітектура, машинобудування та будівництво [10].

Кожна з перерахованих програм є простою для опанування навіть для користувача без досвіду, а також вони дають змогу швидко і якісно створювати програмне забезпечення необхідного вигляду без зайвих витрат часу та ресурсів розробника. Також, дані програми були обрані через можливість отримати доступ до їх повного функціоналу безкоштовно (у випадку з SketchUp і Adobe Photoshop – тимчасовий пробний період).

**2.2 Проектування моделі музею**

Першим кроком проектування буде робота з програмними засобами SketchUp. У порівнянні з багатьма іншими популярними пакетами, цей має низку функцій, позиціонованих авторами як переваги. Головна особливість – майже повна відсутність попередньо встановлених налаштувань. Усі геометричні характеристики під час або відразу після закінчення інструменту встановлюються з клавіатури в полі керування значенням, яке знаходиться в правому нижньому куті робочої області, праворуч від мітки «Вимірювання». Ще однією ключовою особливістю є інструмент Push/Pull («Штовхай/Тягни»), який дозволяє «відтягувати» вбік будь-яку площину, створюючи нові бічні стінки під час руху. Стверджується, що інструмент запатентований. Ви можете наблизити площину до наперед визначеної кривої за допомогою спеціального інструменту Follow Me («Ведення»).

Починаємо роботу з того, що створюємо «фундамент» будівлі, створюючи поверхню, у яку можна буде вписати наш об’єкт у масштабі 1:1. Далі, потрібно накласти на отриману поверхню креслення музею (Рисунок 2.1), а саме вид згори, – це буде нашою заготовкою для дотримання правильних форми та масштабу.

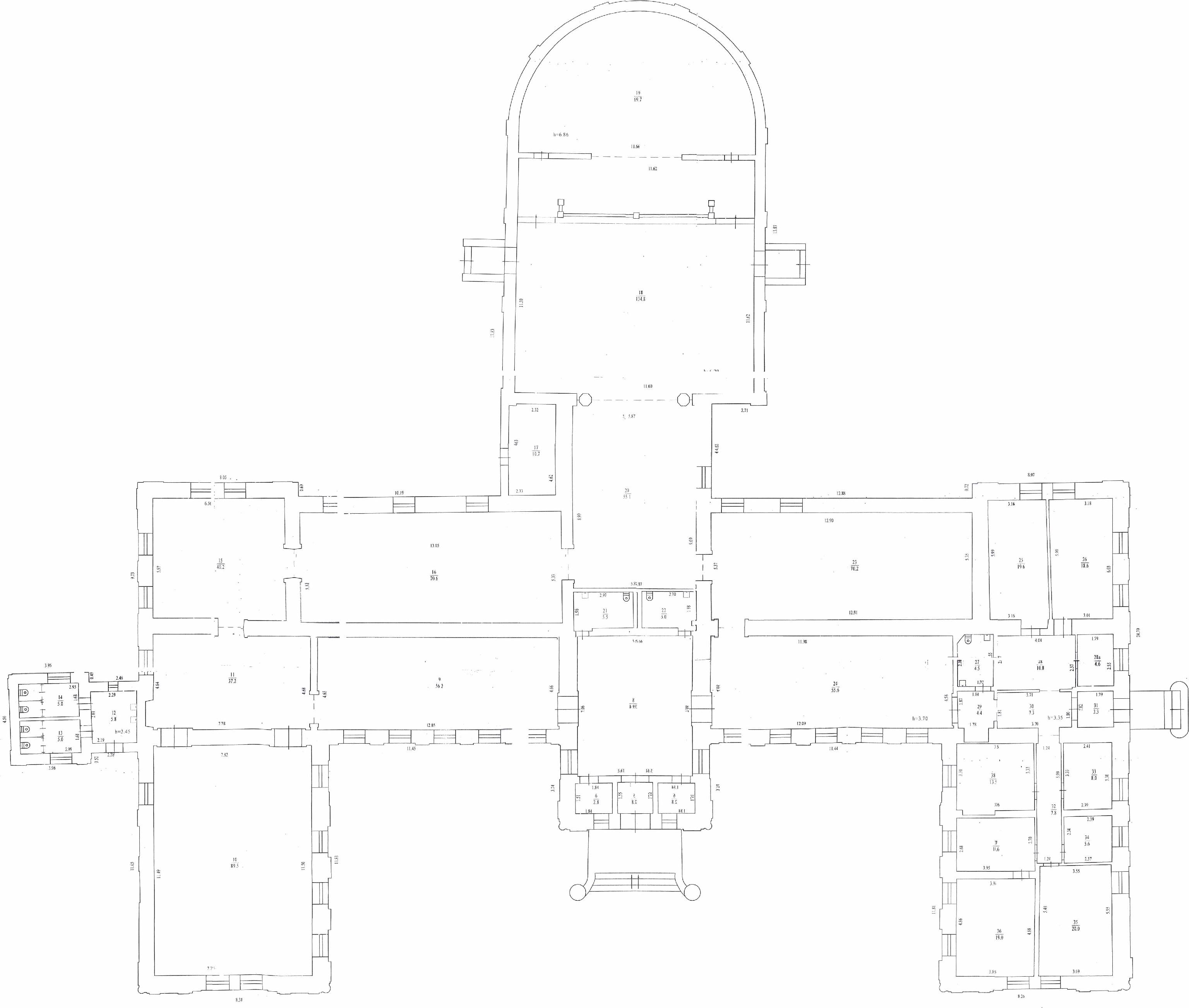


Рисунок 2.1 – Креслення будівлі музею

Після цього починається тривалий та складний процес створення стін шляхом виділення границь на плані та витягування їх за допомогою інструменту «Push/Pull». Складні повторювані об’єкти, такі як вікна, колони, ліпнина, ліхтарі і т.д. створюються як окремі об’єкти та «нарощуються» на стіни, тим самим створюючи реалістичний вигляд моделі. Внутрішня частина будівлі залишається порожньою – її ми будемо створювати далі. Результатом роботи стане модель, показана нижче (Рисунок 2.2 – 2.3)

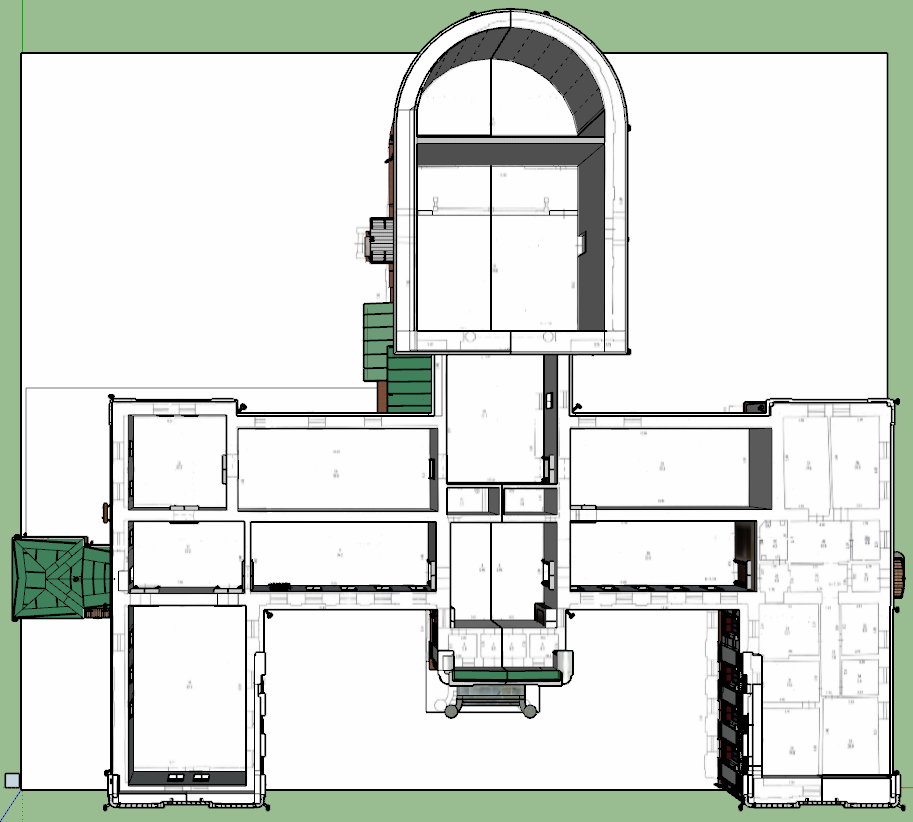


Рисунок 2.2 – Вид на модель згори (можемо бачити вищезгадані креслення)

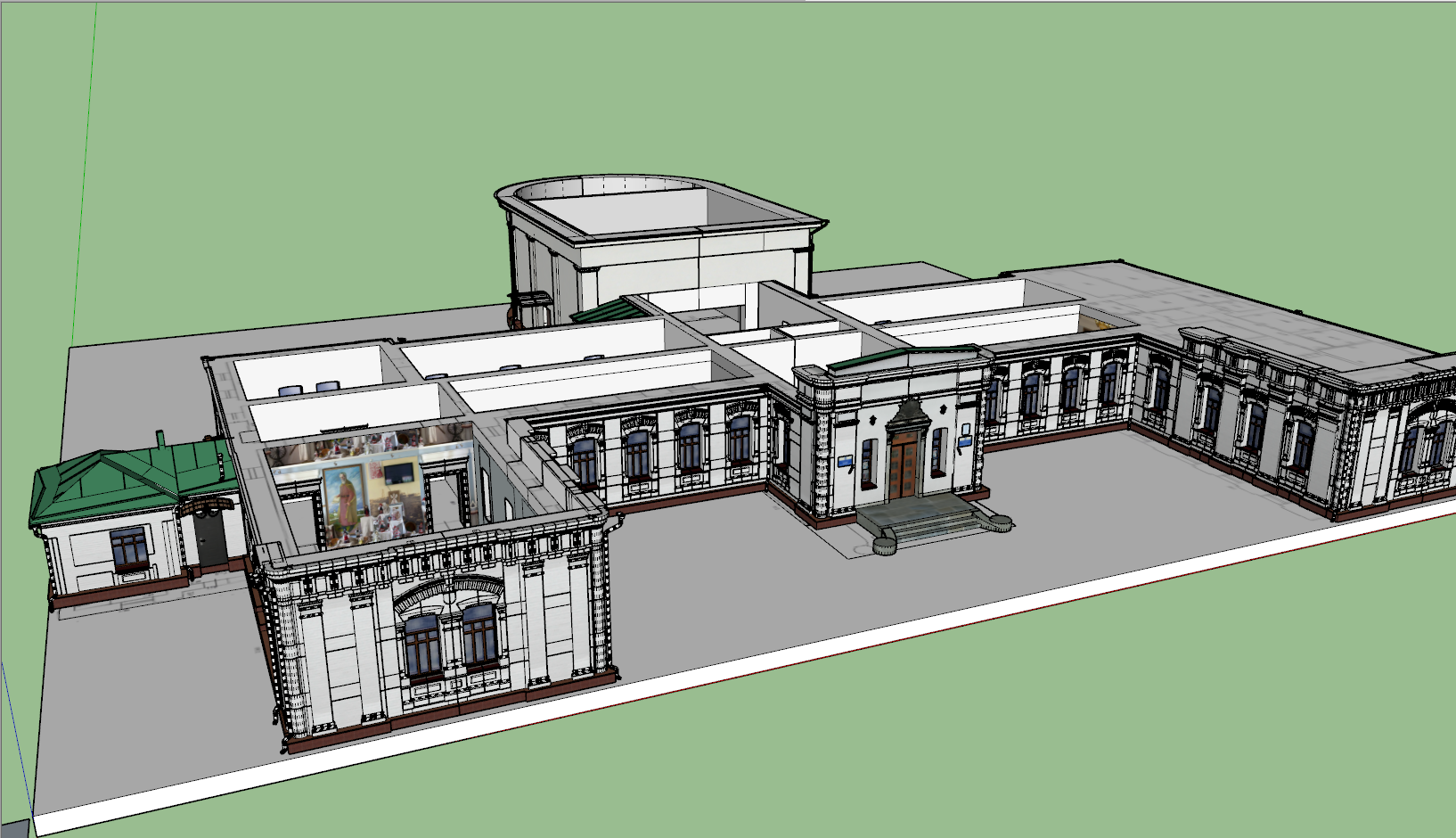


Рисунок 2.3 – Вид на модель збоку (видно як виконано стіни та інші об’єкти)

Як можна бачити з рисунків, будівлю музею складається з вестибюлю, дев’яти експозиційних залів та приміщення для персоналу (на моделі – праворуч). Музей є центральною частиною державного заповідного комплексу «Поле Полтавської битви», загальна площа якого складає 720 гектарів та включає в себе наступні об’єкти, зображені на рисунку 2.4:

1. Братська могила воїнів московської армії
2. Сампсоніївська церква
3. Музей історії Полтавської битви та пам’ятник Петру I
4. Будинок першого музею історії Полтавської битви
5. Ротонда вшанування пам’яті полеглих у Полтавській битві
6. Система редутів московської армії
7. Другий укріплений табір московської армії
8. Командний пункт Петра I
9. Пам’ятник шведам від росіян
10. Пам’ятник шведам від співвітчизників
11. Пам’ятник на місці переправи московської армії через р. Ворсклу
12. Місце переправи ординських військ Темір-Кутлуя через р. Ворсклу влітку 1399 року
13. Місце битви військ гетьмана І. Виговського та полтавського полковника М. Пушкаря влітку 1658 року

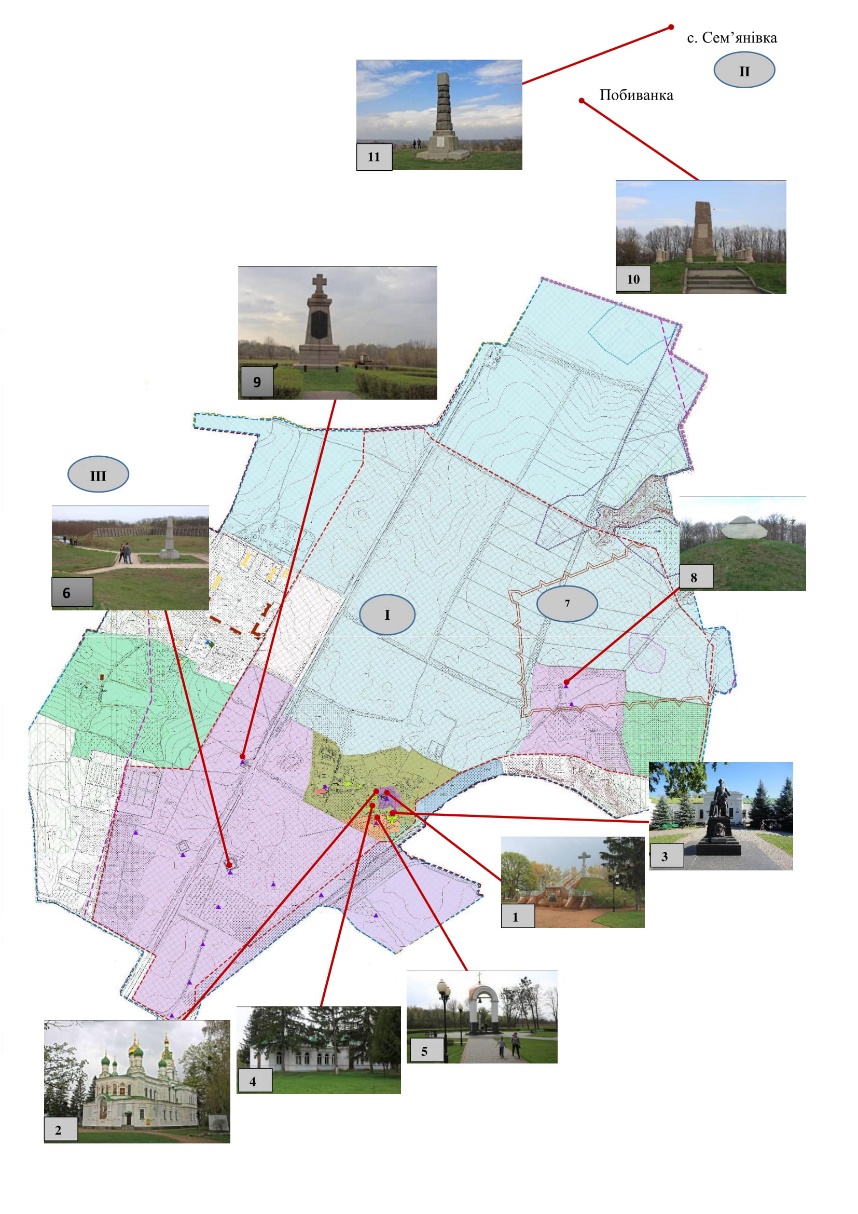


Рисунок 2.4 – Карта заповіднику «Поле Полтавської битви», створена автором роботи на основі матеріалів, наданих співробітниками музею

Дана схема знадобиться нам, аби повністю розуміти масштаби комплексу, хоч у рамках даного проекту нами буде створено лише модель приміщення музею. Проте у майбутньому планується можливий проект віртуалізації всього заповідного комплексу, основою для якого може стати дана робота.

**2.3 Проектування залів музею**

Модель готова і може бути використана для подальшого проектування. Також вона є дуже важливою для подальшої модернізації додатку та підвищення його якості у майбутньому. Зараз же для нас має ключове значення лише внутрішня частина об’єкту, адже основна цінність музею у його наповненні. Тому наступним кроком проектування віртуального музею буде створення залів. Для цього нам доведеться зробити певну кількість фото кожного приміщення музею та провести певні виміри, аби мати змогу створити підходящі фотографічні текстури та об’єкти.

Найбільш правильним напрямком роботи буде створення кожного окремого залу як унікального об’єкту шляхом виділення їх у окремий робочий простір. Після цього необхідно відтворити форму усіх полиць та випуклих об’єктів, що є частиною приміщення, та, в ідеалі, створити кожен окремий експонат у вигляді окремого об’єкта. Далі зал заповнюється створеними об’єктами, створюючи реалістичний зовнішній вигляд. Ми можемо бачити результат такого проектування на прикладі 6-го та 7-го залів музею Полтавської битви, які також називають «діорамою», представлених на рисунку 2.5.

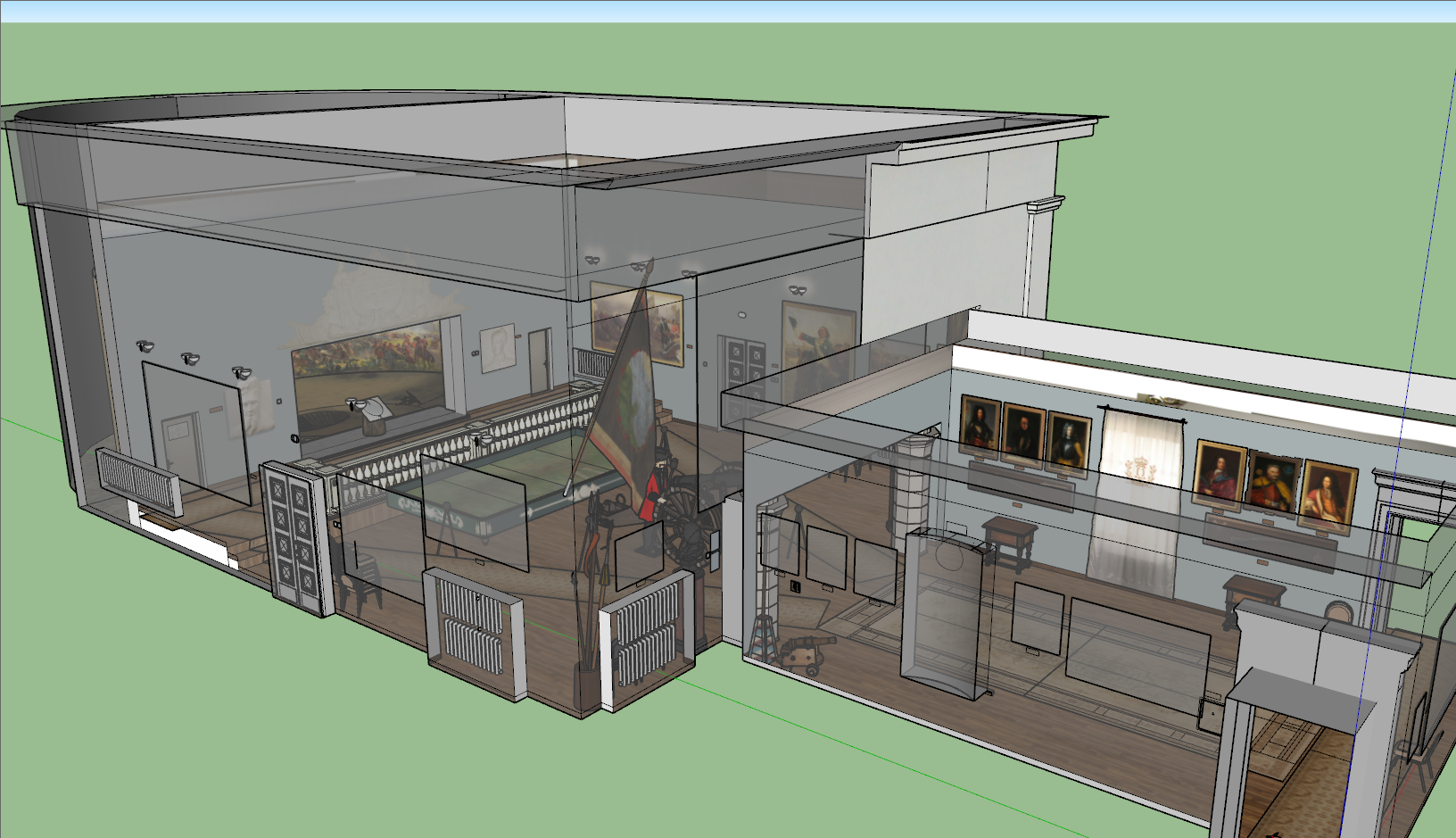


Рисунок 2.5 – Шоста та сьома зали музею як зображення ідеального результату проектування

На рисунку можна побачити, що обидва зали наповнені реалістичними експонатами, кожен з яких є окремим незалежним об’єктом. Дані зали є еталонами та найбільш вірогідними варіантами для подальшого покращення якості віртуального музею та його повноцінного завершення.

Задля спрощення процесу роботи можливо також реалізувати інший метод. Ми не будемо створювати кожен експонат як об’єкт, але відтворимо інтер’єр зали, після чого використаємо фотографічні текстури, заздалегідь створені та оброблені програмними засобами Adobe Photoshop. Створюються такі текстури шляхом «зліплення» декількох фото в одне, після чого отримане зображення завантажується у модель SketchUp і розтягується по стінах та об’ємних деталях приміщення. Результат такого методу роботи можемо бачити на прикладі 5-ї зали, що носить назву «фортеця», яка зображена на рисунку 2.6.

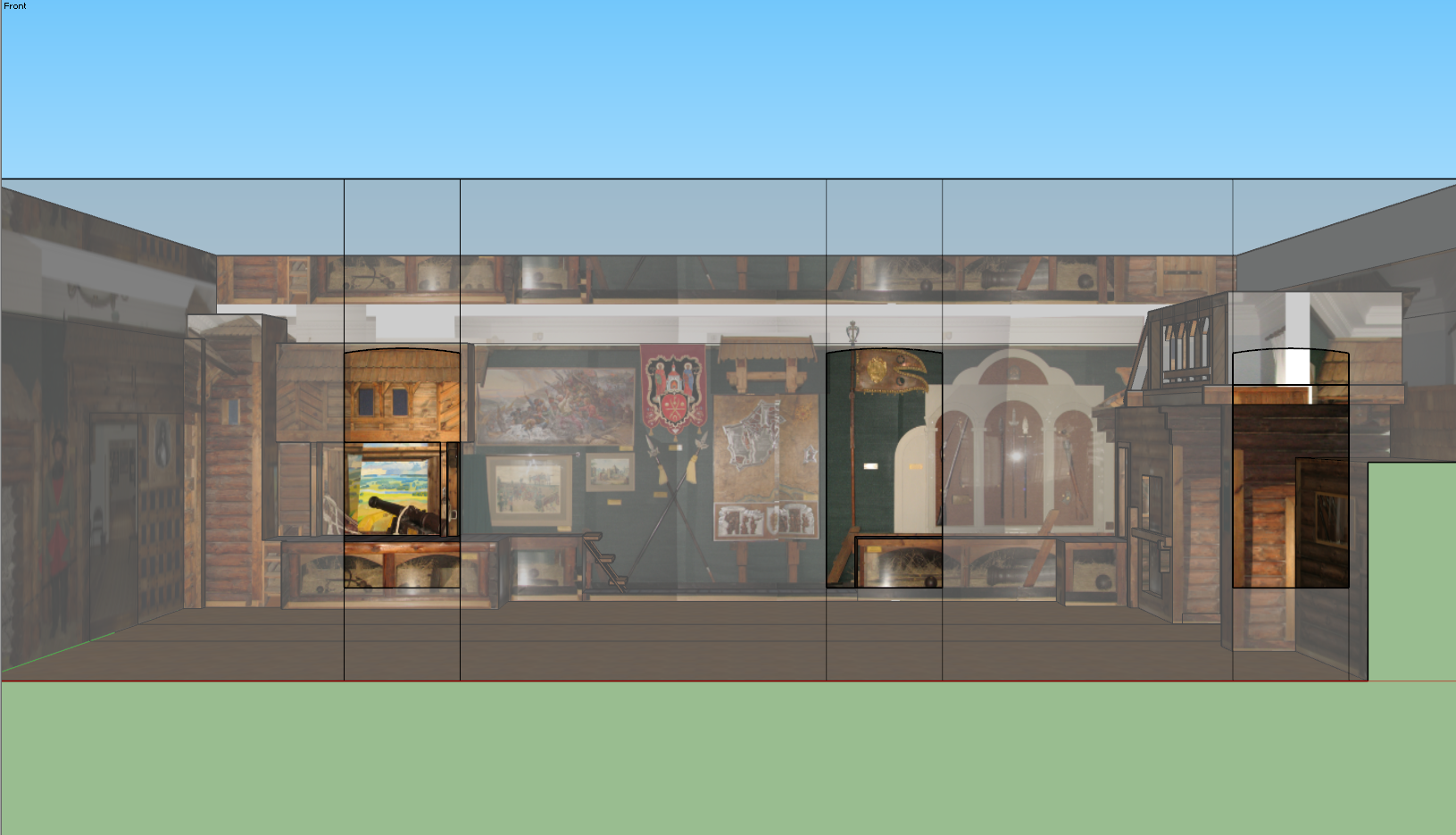


Рисунок 2.6 – П’ята зала музею як приклад спрощеного моделювання

Як бачимо, у залі присутні випуклі частини, які повторюють зовнішній вигляд реальних частин експозиції, таких як дерев’яні стіни фортеці. Таке моделювання дає змогу при значно менших затратах часу та ресурсів відтворити зовнішній вигляд приміщення якого більш наближено до реального.

Однак даний спосіб проектування надто трудомісткий та вимагає забагато часу як для дипломної роботи. Саме тому у рамках даної роботи будуть використані п’ята, шоста та сьома зали, створені заздалегідь у рамках проекту віртуалізації музею історії Полтавської битви, ініціалізованого Лебединським Олегом Валентиновичем, у співробітництві з яким і було створено загальну модель музею та дані зали. З його дозволу напрацювання віртуалізації використано у нашій роботі. Також, даний проект планується продовжити та завершити у такому вигляді, як вищевказані зали, решту приміщень музею.

Надалі для пришвидшення процесу проектування та завершення моделі будемо використовувати найпростіший спосіб, що не потребує великих затрат часу, якісного обладнання та тривалої обробки фотографій для створення текстур. Використовуючи готові фото усіх залів, «накладаємо» їх на стіни залів, створюючи мінімальний ефект об’єму та присутності. Кінцевий результат такого методу моделювання буде мати приблизно такий вигляд, як 1-а зала музею, зображена на рисунку 2.7.



Рисунок 2.7 – Перша зала музею, відтворена за допомогою розміщення оброблених фото на стінах приміщення та/або під правильними кутами

Слідуючи даній методиці, закінчуємо проектування решти залів та виходимо на фінальний етап моделювання віртуальної моделі музею – додаємо п’яту, шосту та сьому зали, які були створені у окремих робочих просторах та збережені у якості компонентів, тобто готових об’єктів, які можна додати до будь-якої моделі у будь-який момент. Рішення додати їх вже при завершенні проектування прийняте з міркувань економії часу та ресурсів комп’ютерного обладнання – перераховані зали у сукупності з рештою 3D-моделі музею помітно знижують швидкість роботи SketchUp, що сповільнює процес моделювання та якість роботи. Результатом нашої роботи буде готова будівля музею, наповнена текстурами та, в деяких залах, об’ємними експонатами.

По закінченню роботи з моделлю нам необхідно експортувати результат у потрібному форматі разом з усіма текстурами, що ми використали для її наповнення. Це знадобиться нам для використання моделі у ігровому рушії для подальшої роботи над проектом [11].

**РОЗДІЛ 3**

**РОЗРОБКА ПРОГРАМНОГО ЗАСТОСУНКУ ВІРТУАЛЬНОГО МУЗЕЮ**

**3.1 Вибір платформи**

Програма розробляється у вигляді, близькому до відеогри, тому для реалізації було обрано потужності ігрового рушія Unity 2019.4 та мови програмування C#.

Синтаксис C# близький до С++ і Java. Мова має строгу статичну типізацію, підтримує поліморфізм, перевантаження операторів, вказівники на функції-члени класів, атрибути, події, властивості, винятки, коментарі у форматі XML. Перейнявши багато від своїх попередників — мов С++, Object Pascal, Модула і Smalltalk — С#, спираючись на практику їхнього використання, виключає деякі моделі, що зарекомендували себе як проблематичні при розробці програмних систем, наприклад, мова С#, на відміну від C++, не передбачає множинне успадкування класів [12].

У рамках ігрового рушія Unity C# використовується для написання скриптів – коротких програм, що задають поведінку об’єктів у програмі. За допомогою них створюються сценарії та команди поведінки, керування переміщенням та багато інших дій, які створюють образ живих об’єктів та формують ігровий процес. З виходом версії 5.2 у 2015 році передбачена вбудована можливість редагувати скрипти у середовищі Visual Studio.

Редактор Unity має інтерфейс, що складається з різних вікон, які можна розташувати на свій розсуд. Завдяки цьому можна проводити налагодження гри чи застосунка прямо в редакторі. Головні вікна — це оглядач ресурсів проекту, інспектор поточного об'єкта, вікно попереднього перегляду, оглядач сцени та оглядач ієрархії ресурсів.

Проект в Unity поділяється на сцени (рівні) — окремі файли, що містять свої ігрові світи зі своїм набором об'єктів, сценаріїв, і налаштувань. Сцени можуть містити в собі як об'єкти-моделі (ландшафт, персонажі, предмети довкілля тощо), так і порожні ігрові об'єкти — ті, що не мають моделі, проте задають поведінку інших об'єктів (тригери подій, точки збереження прогресу тощо). Їх дозволяється розташовувати, обертати, масштабувати, застосовувати до них скрипти. В них є назва (в Unity допускається наявність двох і більше об'єктів з однаковими назвами), може бути тег (мітка) і шар, на якому він повинен відображатися. Так, у будь-якого предмета на сцені обов'язково наявний компонент Transform — він зберігає в собі координати місця розташування, повороту і розмірів по всіх трьох осях. У об'єктів з видимою геометрією також за умовчанням присутній компонент Mesh Renderer, що робить модель видимою. Різні моделі можуть об'єднуватися в набори (ассети) для швидкого доступу до них. Наприклад, моделі споруд на спільну тему.

Unity підтримує фізику твердих тіл і тканини, фізику типу Ragdoll (ганчіркова лялька). У редакторі є система успадкування об'єктів; дочірні об'єкти будуть повторювати всі зміни позиції, повороту і масштабу батьківського об'єкта. Скрипти в редакторі прикріплюються до об'єктів у вигляді окремих компонентів.

У 2D іграх Unity переважно використовує спрайти. В 3D іграх Unity здебільшого використовує тривимірні моделі (меші), на які накладаються текстури (зумовлюють вигляд поверхні об'єктів), матеріали (зумовлюють як поверхня реагуватиме на різні фактори) та шейдери (невеликі скрипти, за яким вираховується зміна кольору кожного пікселя згідно заданих параметрів, як-от розсіяння відбитого світла). В обох видах застосовуються системи часток для відображення субстанцій, таких як рідини чи дим.

Unity підтримує стиснення текстур, міпмапінг і різні налаштування роздільності екрана для кожної платформи; забезпечує бамп-мапінг, мапінг відображень, паралакс-мапінг, затінення навколишнього світла у екранному просторі, динамічні тіні за картами тіней, рендер у текстуру та повноекранні ефекти обробки зображення, такі як зернистість, глибина чіткості, розмиття в русі, відблиски віртуальних лінз або ореол навколо джерел світла [13].

Рендеринг зображення відбувається через віртуальну камеру огляду. В робочій області редактора ігрова сцена може розміщуватися як завгодно, а при рендерингу — так, як її видно з камери. В сцені може бути декілька камер, які рухаються за персонажем чи за вказаною траєкторією. Вигляд з камери подається в двовимірно чи тривимірно (в перспективі або ортографічно). Фон сцени, видимий через камеру, типово зображає небо, утворене скайбоксом, але може презентувати й інше довкілля.

Графічний рушій використовує DirectX (Windows), OpenGL (Mac, Windows, Linux), OpenGL ES (Android, iOS), та спеціальне власне API для Wii. Також підтримуються bump mapping, reflection mapping, parallax mapping, screen space ambient occlusion (SSAO), динамічні тіні з використанням shadow maps, render-to-texture та повноекранні ефекти post-processing.

Unity підтримує файли 3ds Max, Maya, Softimage, Blender, modo, ZBrush, Cinema 4D, Cheetah3D, Adobe Photoshop, Adobe Fireworks та Allegorithmic Substance. В ігровий проєкт Unity можна імпортувати об'єкти цих програм та виконувати налаштовування за допомогою графічного інтерфейсу

Для написання шейдерів використовується ShaderLab, що підтримує шейдерні програми написані на GLSL або Cg. Шейдер може включати декілька варіантів реалізації, що дозволяє Unity визначати найкращий варіант для конкретної відеокарти. Unity також має вбудовану підтримку фізичного рушія Nvidia PhysX (колишнього Ageia), підтримку симуляції одягу в системі реального часу на довільній та прив'язаній полігональній сітці (починаючи з Unity 3.0), підтримку системи ray casts та шарів зіткнення.

Виходячи з усього вищесказаного, Unity ідеально підходить для створення нашого програмного застосунку у вигляді гри на основі створеної у попередньому розділі тривимірної моделі музею історії Полтавської битви.

**3.2 Постановка сцени**

Перш ніж розпочати роботу над поведінкою об’єктів, їх необхідно правильно додати до нашого проекту. Аби це зробити, слід експортувати модель музею у підходящому форматі. У нашому випадку найкращим вибором буде формат .fbx, доступний у SketchUp Pro та його пробній версії, якою ми і користуємося. Даний формат дозволяє вивантажити повну геометрію об’єкту та зберегти усі текстури у вигляді, доступному для подальшого використання ігровим рушієм.

Завершивши експорт моделі, нам потрібно імпортувати її в наш проект. Для цього відриваємо наш проект, натискаємо правою клавішею миші на вікно у нижній частині екрану (Assets), та створюємо папку “Textures” (Текстури). Після цього відкриваємо її, знову відкриваємо контекстне меню та шукаємо вкладку “Import New Asset…” (Рисунок 3.1) та через провідник обираємо всі експортовані текстури нашої моделі. Вони знадобляться нам, аби модель додалася до сцени у тому ж вигляді, у якому ми її створили. Після завантаження текстур нам слід повернутися назад у папку “Assets” та імпортувати туди таким же чином саму модель музею. Якщо всі пункти виконані правильно, ми зможемо додати до сцени модель у вигляді ігрового об’єкта та використовувати її у якості ігрової локації.

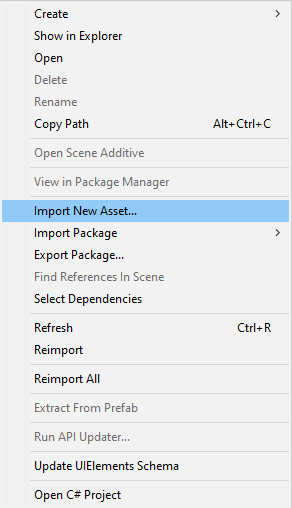


Рисунок 3.1 – Контекстне меню проекту Unity

Наступним кроком буде розміщення локації на самій сцені. Для цього достатньо перетягнути імпортований музей до вікна ієрархії об’єктів і він з’явиться у вікні сцени. Так як ми заздалегідь імпортували усі текстури до нашого проекту, модель буде вставлена у тому ж вигляді, який ми створили засобами SketchUp. Як бачимо з рисунку 3.2, все було виконано правильно, тому ми бачимо повноцінну модель у потрібному вигляді.



Рисунок 3.2 – Завантажена модель музею у вікні редактора Unity

Також для покращення зовнішнього вигляду сцени в Unity передбачене налаштування направленого світла та інших джерел освітлення, проте в рамках нашого проекту нам вистачить і стандартного, тому опис його налаштування ми пропустимо.

**3.3 Робота з камерою**

Не менш важливим за будівлю музею для нашого проекту є ігровий об’єкт камери, за допомогою якого ми і матимемо змогу оглядати будівлю з усіх боків. Камера спрямована вниз по осі Z ігрового об’єкту і поводиться так само, як і всі інші об’єкти. У режимі перегляду “Сцена” ви можете побачити лінії, що представляють собою усічену піраміду камери. Усічений конус являє собою тверду форму, схожу на піраміду, вершина якої зрізана паралельно основі. Це форма області, яку можна побачити і візуалізувати за допомогою камери у режимі перспективи.

Коли ви робите гру, у вас є кілька варіантів, як зробити, щоб камера рухалася саме так, як вам потрібно [14]. Для цього є два основних рішення:

* Написати скрипт поведінки камери;
* Скористатися вбудованим у ігровий рушій Unity засобом, що називається Cinemachine.

Ми скористаємося першим варіантом і напишемо невеликий скрипт мовою програмування C#, який дозволить нам керувати камерою за власним розсудом. Аби почати роботу зі скриптом, знову викликаємо контекстне меню у вкладці Assets, створюємо нову папку “Scripts” (для зручнішого керування імпортованими об’єктами) і в ній обираємо пункт “Create – C# Script”.

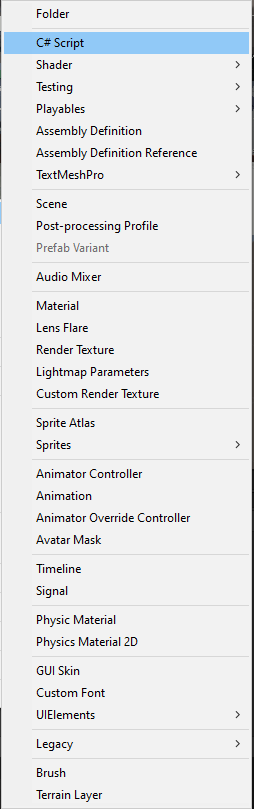


Рисунок 3.3 – Контекстне меню створення об’єктів (частково)

Створюється порожній сценарій, що має базовий набір команд та два класи, що є базовими для будь-якого скрипта. Відкриваємо створений файл для редагування у вікні Visual Studio. Для нашого проекту нам необхідно налаштувати керування камерою таким чином, аби вона могла переміщуватися у трьох вимірах, підніматися вгору та опускатися вниз, а також обертатися, коли користувач затискає праву клавішу миші та рухає мишкою.

Враховуючи перераховані вимоги, пишемо скрипт, який буде зчитувати введення даних з клавіатури та миші та перетворювати їх на вектор руху. Перш за все, створюємо публічний клас, який буде реагувати на праву клавішу миші та рух і транслювати це у зміну положення поля зору. Реалізується дане зчитування наступною частиною коду:

public void SetFromTransform(Transform t)

{

pitch = t.eulerAngles.x;

yaw = t.eulerAngles.y;

roll = t.eulerAngles.z;

x = t.position.x;

y = t.position.y;

z = t.position.z;

}

public void Translate(Vector3 translation)

{

Vector3 rotatedTranslation = Quaternion.Euler(pitch, yaw, roll) \* translation;

x += rotatedTranslation.x;

y += rotatedTranslation.y;

z += rotatedTranslation.z;

}

public void LerpTowards(CameraState target, float positionLerpPct, float rotationLerpPct)

{

yaw = Mathf.Lerp(yaw, target.yaw, rotationLerpPct);

pitch = Mathf.Lerp(pitch, target.pitch, rotationLerpPct);

roll = Mathf.Lerp(roll, target.roll, rotationLerpPct);

x = Mathf.Lerp(x, target.x, positionLerpPct);

y = Mathf.Lerp(y, target.y, positionLerpPct);

z = Mathf.Lerp(z, target.z, positionLerpPct);

}

public void UpdateTransform(Transform t)

{

t.eulerAngles = new Vector3(pitch, yaw, roll);

t.position = new Vector3(x, y, z);

}

Реалізувавши та налаштувавши обертання камери, тепер нам необхідно провести зчитування натискання клавіш руху, які будуть переміщувати камеру у тривимірному просторі. Для відтворення даної функції буде використано наступний набір клавіш:

* W – рух уперед;
* S – рух назад;
* A – рух ліворуч;
* D – рух праворуч;
* Q – рух вниз;
* E – рух вгору;

Для створення такої поведінки у Unity та C# передбачена можливість зчитування векторів осей X, Y та Z. Перераховані клавіші є стандартними у більшості ігор, тому їх призначення не потребує великої кількості коду і виконується наступним чином:

Vector3 GetInputTranslationDirection()

{

Vector3 direction = new Vector3();

if (Input.GetKey(KeyCode.W))

{

direction += Vector3.forward;

}

if (Input.GetKey(KeyCode.S))

{

direction += Vector3.back;

}

if (Input.GetKey(KeyCode.A))

{

direction += Vector3.left;

}

if (Input.GetKey(KeyCode.D))

{

direction += Vector3.right;

}

if (Input.GetKey(KeyCode.Q))

{

direction += Vector3.down;

}

if (Input.GetKey(KeyCode.E))

{

direction += Vector3.up;

}

return direction;

}

Таким чином ми зможемо керувати полем зору та рухом камери за власним бажанням та вільно переміщуватися навколо об’єкту. На даному етапі не обов’язково обмежувати можливість руху камери, тому у користувача зберігається можливість проходити крізь стіни, але це не є проблемою. Вирішення цього питання залишимо на проект модернізації даної програми.

Надалі нам необхідно зробити так, щоб програма зчитувала дії користувача постійно у реальному часі. Для цього у кожному скрипті передбачений клас Update, що дозволяє виконувати якісь повторювані дії кожен момент часу. Він найкраще підходить для зчитування натискання клавіш та руху мишею. Реалізуємо бажану поведінку щосекундного виявлення зміни значення векторів, до яких прив’язані обрані нами клавіші, за допомогою наступної кодової частини:

if (Input.GetKey(KeyCode.Escape))

{

Application.Quit();

#if UNITY\_EDITOR

UnityEditor.EditorApplication.isPlaying = false;

#endif

}

// Hide and lock cursor when right mouse button pressed

if (Input.GetMouseButtonDown(1))

{

Cursor.lockState = CursorLockMode.Locked;

}

// Unlock and show cursor when right mouse button released

if (Input.GetMouseButtonUp(1))

{

Cursor.visible = true;

Cursor.lockState = CursorLockMode.None;

}

// Rotation

if (Input.GetMouseButton(1))

{

var mouseMovement = new Vector2(Input.GetAxis("Mouse X"), Input.GetAxis("Mouse Y") \* (invertY ? 1 : -1));

var mouseSensitivityFactor = mouseSensitivityCurve.Evaluate(mouseMovement.magnitude);

m\_TargetCameraState.yaw += mouseMovement.x \* mouseSensitivityFactor;

m\_TargetCameraState.pitch += mouseMovement.y \* mouseSensitivityFactor;

}

// Translation

var translation = GetInputTranslationDirection() \* Time.deltaTime;

// Speed up movement when shift key held

if (Input.GetKey(KeyCode.LeftShift))

{

translation \*= 10.0f;

}

Повний текст даного скрипта наведено у додатку А.

Тепер, коли ми маємо готовий сценарій поведінки об’єкта, потрібно розмістити наш об’єкт, тобто камеру, у потрібному стартовому місці та додати до неї скрипт у вигляді компоненту у вікні Інспектора об’єктів. Зробити це можна обравши камеру у вікні Ієрархії та перетягнувши сценарій у вікно Інспектора (Рисунок 3.4). Тепер наша камера здатна вільно переміщуватися сценою.

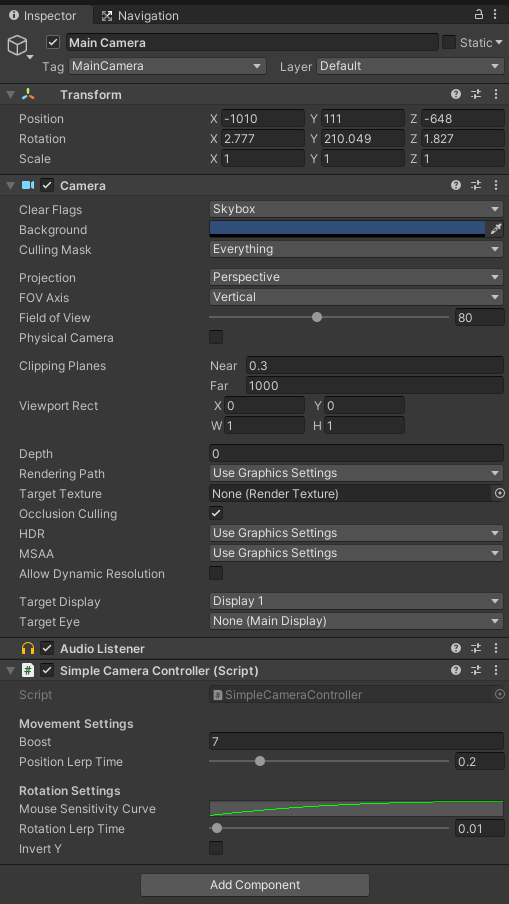


Рисунок 3.4 – Вікно Інспектора об’єктів камери

Отже, ми додали до нашого застосунку камеру, налаштували її положення та створили сценарій поведінки. На даному етапі вже можна казати, що проект готовий до використання. Далі нам залишається тільки завершити деякі налаштування та перейти на завершальний етап розробки.

**ВИСНОВКИ**

У даній роботі було спроектовано і створено програмний застосунок віртуального музею “Поле Полтавської битви”, який дає змогу користувачу розглянути будівлю музею з будь-якого ракурсу та роздивитися зовнішній вигляд залів, характеризується зручністю та простотою у використанні, дає можливість вільного переміщення об’єктом та має просту програмну частину.

У першому розділі було проаналізовано існуючі методи інтеграції сучасних технологій у сфері туризму. Крім того, розглянуто варіанти та методи створення віртуальних музеїв, оглянуто яскравих представників цієї технології та визначено найбільш оптимальний спосіб розробки проекту у нашому випадку. Було зроблено висновок, що надбання сучасного технологічного прогресу активно використовуються робітниками культури, в тому числі музеями. Визначено, що технологія віртуального музею набуває популярності та впроваджується у все більшій кількості закладів.

Другий розділ присвячено плануванню та проектуванню тривимірної моделі музею Полтавської битви. Також у даному розділі проведено наповнення залів текстурами та об’ємними об’єктами.

У третьому розділі представлено детальний опис розробки програмного продукту, а саме застосунку на базі ігрового рушія Unity. У якості ігрової локації використано створену у другому розділі тривимірну модель музею, а переміщення довкола неї реалізовано вільною камерою, керування якою здійснюється завдяки сценарію поведінки, написаному мовою програмування C# програмними засобами Visual Studio у якості IDE.

Запропонований програмний застосунок пройшов випробування і рекомендований до промислового впровадження

Заявлене програмне рішення може бути використане у галузі культури та туризму, зокрема для популяризації музею Полтавської битви як пам’ятки культури та для заохочення туристів до його відвідування.

**СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ**

1. Структурна та формальна моделі віртуального музею [Електронний ресурс] / П.І. Жежнич, Ю.В. Ришковець. – Львів, Нац. ун-т “Львівська політехніка”; 2009. – Режим доступу до статті: <http://ena.lp.edu.ua/bitstream/ntb/719/1/12.pdf>
2. ДІКЗ “Поле Полтавської битви” [Електронний ресурс] – Режим доступу: <https://battle-poltava.org/>
3. Tokyo National Museum [Електронний ресурс] – Режим доступу: <https://www.tnm.jp/?lang=en>
4. Государственный Эрмитаж [Електронний ресурс] – Режим доступу: <https://hermitagemuseum.org/wps/portal/hermitage?lng=ru>
5. Smithsonian national Museum of Natural History [Електронний ресурс] – Режим доступу: <https://naturalhistory2.si.edu/vt3/NMNH/z_tour-022.html>
6. Музеї України просто неба [Електронний ресурс] – Режим доступу: <https://museums.authenticukraine.com.ua/ua/>
7. Музей-заповідник “Жовківський замок” [Електронний ресурс] – Режим доступу: <http://lvivgallery.org.ua/museums/muzey-zapovidnyk-zhovkivskyy-zamok>
8. Adobe Photoshop [Електронний ресурс] – Режим доступу: <https://www.adobe.com/ua/products/photoshop.html?sdid=B4XQ3XKB&mv=search&skwcid=AL!3085!3!442252953561!e!!g!!adobe%20photoshop&ef_id=CjwKCAjwt8uGBhBAEiwAayu_9ZaSwYmjydjyrgcRXfYqCj9Ph54Ge0DE1lAN7Xz51FMHgaRpGZj-7RoCEjAQAvD_BwE:G:s&s_kwcid=AL!3085!3!442252953561!e!!g!!adobe%20photoshop!1712281352!67017686636&gclid=CjwKCAjwt8uGBhBAEiwAayu_9ZaSwYmjydjyrgcRXfYqCj9Ph54Ge0DE1lAN7Xz51FMHgaRpGZj-7RoCEjAQAvD_BwE>
9. SketchUp [Електронний ресурс] – Режим доступу: <https://www.sketchup.com/>
10. Unity, платформа розробки у реальному часі [Електронний ресурс] – Режим доступу: <https://unity.com/ru>
11. How to - Import a Sketchup (.skp) Model into Unity (with Textures) [Електронний ресурс] / RaventurnPatrick. – 2011. – Режим доступу: <https://forum.unity.com/threads/how-to-import-a-sketchup-skp-model-into-unity-with-textures.99965/>
12. C Sharp (programming language) [Електронний ресурс] – Режим доступу: <https://en.wikipedia.org/wiki/C_Sharp_(programming_language)>
13. Unity (game engine) [Електронний ресурс] – Режим доступу: <https://en.wikipedia.org/wiki/Unity_(game_engine)>
14. Unity Forums [Електронний ресурс] – Режим доступу: <https://forum.unity.com/>

**ДОДАТОК А**

**ТЕКСТ СЦЕНАРІЮ ПОВЕДІНКИ КАМЕРИ**

using UnityEngine;

namespace UnityTemplateProjects

{

public class SimpleCameraController : MonoBehaviour

{

class CameraState

{

public float yaw;

public float pitch;

public float roll;

public float x;

public float y;

public float z;

public void SetFromTransform(Transform t)

{

pitch = t.eulerAngles.x;

yaw = t.eulerAngles.y;

roll = t.eulerAngles.z;

x = t.position.x;

y = t.position.y;

z = t.position.z;

}

public void Translate(Vector3 translation)

{

Vector3 rotatedTranslation = Quaternion.Euler(pitch, yaw, roll) \* translation;

x += rotatedTranslation.x;

y += rotatedTranslation.y;

z += rotatedTranslation.z;

}

public void LerpTowards(CameraState target, float positionLerpPct, float rotationLerpPct)

{

yaw = Mathf.Lerp(yaw, target.yaw, rotationLerpPct);

pitch = Mathf.Lerp(pitch, target.pitch, rotationLerpPct);

roll = Mathf.Lerp(roll, target.roll, rotationLerpPct);

x = Mathf.Lerp(x, target.x, positionLerpPct);

y = Mathf.Lerp(y, target.y, positionLerpPct);

z = Mathf.Lerp(z, target.z, positionLerpPct);

}

public void UpdateTransform(Transform t)

{

t.eulerAngles = new Vector3(pitch, yaw, roll);

t.position = new Vector3(x, y, z);

}

}

CameraState m\_TargetCameraState = new CameraState();

CameraState m\_InterpolatingCameraState = new CameraState();

[Header("Movement Settings")]

[Tooltip("Exponential boost factor on translation, controllable by mouse wheel.")]

public float boost = 3.5f;

[Tooltip("Time it takes to interpolate camera position 99% of the way to the target."), Range(0.001f, 1f)]

public float positionLerpTime = 0.2f;

[Header("Rotation Settings")]

[Tooltip("X = Change in mouse position.\nY = Multiplicative factor for camera rotation.")]

public AnimationCurve mouseSensitivityCurve = new AnimationCurve(new Keyframe(0f, 0.5f, 0f, 5f), new Keyframe(1f, 2.5f, 0f, 0f));

[Tooltip("Time it takes to interpolate camera rotation 99% of the way to the target."), Range(0.001f, 1f)]

public float rotationLerpTime = 0.01f;

[Tooltip("Whether or not to invert our Y axis for mouse input to rotation.")]

public bool invertY = false;

void OnEnable()

{

m\_TargetCameraState.SetFromTransform(transform);

m\_InterpolatingCameraState.SetFromTransform(transform);

}

Vector3 GetInputTranslationDirection()

{

Vector3 direction = new Vector3();

if (Input.GetKey(KeyCode.W))

{

direction += Vector3.forward;

}

if (Input.GetKey(KeyCode.S))

{

direction += Vector3.back;

}

if (Input.GetKey(KeyCode.A))

{

direction += Vector3.left;

}

if (Input.GetKey(KeyCode.D))

{

direction += Vector3.right;

}

if (Input.GetKey(KeyCode.Q))

{

direction += Vector3.down;

}

if (Input.GetKey(KeyCode.E))

{

direction += Vector3.up;

}

return direction;

}

void Update()

{

// Exit Sample

if (Input.GetKey(KeyCode.Escape))

{

Application.Quit();

#if UNITY\_EDITOR

UnityEditor.EditorApplication.isPlaying = false;

#endif

}

// Hide and lock cursor when right mouse button pressed

if (Input.GetMouseButtonDown(1))

{

Cursor.lockState = CursorLockMode.Locked;

}

// Unlock and show cursor when right mouse button released

if (Input.GetMouseButtonUp(1))

{

Cursor.visible = true;

Cursor.lockState = CursorLockMode.None;

}

// Rotation

if (Input.GetMouseButton(1))

{

var mouseMovement = new Vector2(Input.GetAxis("Mouse X"), Input.GetAxis("Mouse Y") \* (invertY ? 1 : -1));

var mouseSensitivityFactor = mouseSensitivityCurve.Evaluate(mouseMovement.magnitude);

m\_TargetCameraState.yaw += mouseMovement.x \* mouseSensitivityFactor;

m\_TargetCameraState.pitch += mouseMovement.y \* mouseSensitivityFactor;

}

// Translation

var translation = GetInputTranslationDirection() \* Time.deltaTime;

// Speed up movement when shift key held

if (Input.GetKey(KeyCode.LeftShift))

{

translation \*= 10.0f;

}

// Modify movement by a boost factor (defined in Inspector and modified in play mode through the mouse scroll wheel)

boost += Input.mouseScrollDelta.y \* 0.2f;

translation \*= Mathf.Pow(2.0f, boost);

m\_TargetCameraState.Translate(translation);

// Framerate-independent interpolation

// Calculate the lerp amount, such that we get 99% of the way to our target in the specified time

var positionLerpPct = 1f - Mathf.Exp((Mathf.Log(1f - 0.99f) / positionLerpTime) \* Time.deltaTime);

var rotationLerpPct = 1f - Mathf.Exp((Mathf.Log(1f - 0.99f) / rotationLerpTime) \* Time.deltaTime);

m\_InterpolatingCameraState.LerpTowards(m\_TargetCameraState, positionLerpPct, rotationLerpPct);

m\_InterpolatingCameraState.UpdateTransform(transform);

}

}

}