

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

**ТЕХНОЛОГІЇ ДОПОВНЕНОЇ РЕАЛЬНОСТІ.  
КОМП'ЮТЕРНИЙ ПРАКТИКУМ**

*Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського  
як навчальний посібник для здобувачів ступеня бакалавра за освітньою  
програмою «Технології друкованих і електронних видань»  
спеціальності 186 Видавництво та поліграфія*

Київ  
КПІ ім. Ігоря Сікорського  
2021

Технології доповненої реальності. Комп'ютерний практикум. [Електронний ресурс] // навч. посіб. для студентів спеціальності 186 «Видавництво та поліграфія» // Укладачі: К. І. Золотухіна, Д. І. Баранова – Електронні текстові дані (1 файл: 1,9 Мбайт). – К.: КПІ ім. Ігоря Сікорського, 2021. – 53 с.

*Гриф надано Методичною радою КПІ ім. Ігоря Сікорського  
(протокол № 1 від 16.09.2021 р.)  
за поданням Вченої ради ВПІ (протокол № 1 від 30.08.2021 р.)*

Електронне мережне навчальне видання

## **ТЕХНОЛОГІЇ ДОПОВНЕНОЇ РЕАЛЬНОСТІ: КОМП'ЮТЕРНИЙ ПРАКТИКУМ**

Укладачі: *Золотухіна Катерина Ігорівна*, канд. техн. наук, доц.  
*Баранова Дарина Іванівна*, асистент

Відповідальний редактор: *Киричок Т. Ю.*, д-р техн. наук, проф.

Рецензент: *Зоренко О. В.*, канд. техн. наук, доцент.

Навчальний посібник відповідає навчальній програмі дисципліни «Технології доповненої реальності» спеціальності 186 «Видавництво та поліграфія» освітньої програми «Технології друкованих і електронних видань» підготовки студентів Видавничо-поліграфічного інституту. Наведено перелік робіт як індивідуальних завдань комп'ютерного практикуму. Показано застосування теоретичного матеріалу до розв'язування поставлених практичних задач у відповідності до лабораторної роботи.

Для студентів ВПІ КПІ ім. Ігоря Сікорського спеціальності 186 Видавництво та поліграфія.

## ЗМІСТ

Передмова.....	4
1. Мета, завдання і тематика робіт комп'ютерного практикуму.....	5
2. Основні вимоги до виконання робіт комп'ютерного практикуму.....	6
3. Зміст та перелік робіт комп'ютерного практикуму.....	7
3.1 Створення елементів доповненої реальності за допомогою онлайн-ресурсу ROAR.....	7
3.2 Створення елементів доповненої реальності за допомогою UNITY..	14
3.3 Створення ефектів за допомогою UNITY та додавання їх до елементів доповненої реальності .....	23
3.4 Створення меню та переходів між сценами додатку з елементами доповненої реальності за допомогою UNITY.....	45
3.5 Розроблення маркерів доповненої реальності та визначення впливу характеристик задрукованого матеріалу на їх поліграфічне відтворення та відтворення на екрані цифрового контенту.....	49
Рекомендований перелік літературних джерел.....	52
Додаток А. Приклад оформлення титульного аркушу.....	53

## ПЕРЕДМОВА

Навчальний посібник охоплює основні поняття, термінологію, закономірності та засоби для створення доповненої реальності, опанування наявних технологій, підготовки мультимедійних компонентів для подальшого їх використання при реалізації технологій доповненої реальності, роботи з бібліотеками доповненої реальності, знаходження найбільш зручної форми подання цього матеріалу з урахуванням його інформаційного рівня, здобуття практичних навичок роботи у спеціальних комп'ютерних програмах.

Навчальний посібник поглибить уявлення про коло професійних питань з технологій доповненої реальності, професійне сприйняття продуктів, мов програмування для розроблення додатків доповненої реальності, розвине індивідуальне творче мислення. Надбані знання допоможуть орієнтуватись у сучасному стані технологій доповненої реальності, допоможуть ефективно вирішувати завдання при створенні доповненої реальності.

Посібник призначено для студентів денної та заочної форми навчання технічних спеціальностей. Його можна використати також для підготовки до занять, заліків, екзаменів студентам всіх форм навчання, які вивчають подібний матеріал. Курс відповідає нагальній ринковій потребі підготовки сучасних фахівців.

## **1. МЕТА, ЗАВДАННЯ І ТЕМАТИКА РОБІТ КОМП'ЮТЕРНОГО ПРАКТИКУМУ**

Мета робіт практикуму полягає в закріпленні знань, одержаних студентами під час вивчення дисципліни «Технології доповненої реальності», застосуванні отриманих знань для вирішення конкретних завдань, сприянні самостійності у аналізі та прийнятті важливих професійних рішень, які б підвищили технічний рівень підготовки. Програма курсу «Технології доповненої реальності» охоплює основні поняття, термінологію, закономірності та засоби для створення доповненої реальності, опанування наявних технологій, підготовки мультимедійних компонентів для подальшого їх використання при реалізації технологій доповненої реальності, роботи з бібліотеками доповненої реальності, знаходження найбільш зручної форми подання цього матеріалу з урахуванням його інформаційного рівня, здобуття практичних навичок роботи у спеціальних комп'ютерних програмах.

Зміст і структура навчального посібнику відображає новітні тенденції розвитку видавничо-поліграфічної справи і забезпечує практичне вирішення завдань при створенні доповненої реальності.

Метою робіт практикуму із даної дисципліни, яка є основоположною при застосуванні технологій доповненої реальності, є навчання майбутнього фахівця умінню здійснювати вибір оптимальних режимів підготовки і проектування мультимедійних компонентів, міток, ідентифікаторів в межах сучасної КВС; здійснювати вибір апаратного та програмного забезпечення КВС для оптимального процесу створення доповненої реальності; застосовувати основний перелік команд програм для створення доповненої реальності; аналізувати доцільність використання команд, знати всі можливі способи їх застосування; навчитися створювати маркери, опрацьовувати мультимедійний матеріал та застосовувати мови програмування для додатків тощо. Роботи комп'ютерного практикуму виконуються студентами першого (бакалаврського) рівня вищої освіти ступеня «бакалавр».

## **2. ОСНОВНІ ВИМОГИ ДО ВИКОНАННЯ РОБІТ КОМП'ЮТЕРНОГО ПРАКТИКУМУ**

Роботи практикуму містять відповідні завдання. При виконанні робіт необхідно дотримуватися наведених нижче правил. Роботи, виконані без дотримання цих правил, можуть бути повернені студенту для доопрацювання.

Протокол практичної роботи оформлюється у вигляді роздрукованих сторінок формату А4, оформлення якої здійснюється із дотриманням вимог ДСТУ 3008-2015.

Типова структура практичної роботи містить:

- титульний аркуш (оформлення у додатку);
- аркуш завдання;
- основна частина;
- додатки (за необхідністю).

Оформлення звіту та порядок захисту

Звіт роботи виконується на аркушах А4, в протоколі стисло відображається хід роботи, отримані результати та висновки. Зміст виконаної роботи ілюструється на електронних носіях та, по можливості, виготовленою продукцією. При захисті студент повинен розуміти зміст роботи, також знати відповіді на запитання щодо можливостей використання основних функцій застосованих програмних засобів.

### 3. ЗМІСТ ТА ПЕРЕЛІК РОБІТ КОМП'ЮТЕРНОГО ПРАКТИКУМУ

#### КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 3.1

#### СТВОРЕННЯ ЕЛЕМЕНТІВ ДОПОВНЕНОЇ РЕАЛЬНОСТІ ЗА ДОПОМОГОЮ ОНЛАЙН-РЕСУРСУ ROAR

**Мета:** Створення об'єктів доповненої реальності за допомогою веб-ресурсу The RoAR

#### Теоретичні відомості

##### 1. Загальні відомості про ресурс

**Доповнена реальність** – технологія, що дозволяє впровадити віртуальні об'єкти (текст, ілюстрації, анімація, аудіо, відео тощо) у реальний світ людини шляхом сканування спеціального об'єкту-маркеру, що роздрукований на будь-якому задрукованому матеріалі чи відтворений на екрані монітору, за допомогою камери смартфона чи планшету чи будь-якого іншого пристрою з камерою та подальшого відтворення цифрового контенту на екрані цього пристрою.

Найбільш популярними видами даної технології є маркерна та безмаркерна. Даний практикум буде присвячений розробці елементу доповненої реальності, що базується на безмаркерній технології. У даному випадку маркер-зображення все таки є. Він являє собою складне деталізоване зображення, проте основою для розпізнавання слугують різноманітні дані – положення користувача у просторі, його геолокація, кут повороту пристрою тощо. Одним із видів програмного забезпечення, що застосовується для створення безмаркерної доповненої реальності є **онлайн-платформи**. Вони дозволяють легко об'єднувати між собою віртуальні об'єкти та самостійно їх зашифровують у елемент доповненої реальності. Вони дають змогу працювати з певним набором безкоштовних функцій та певною кількістю «маркерів». Проте для більш деталізованого додатку слід активувати платну

версію. До таких додатків можна віднести **BlippAR, Vuforia, RoAR, PlugXR** тощо. У даному практикумі будуть розглянуті принципи роботи на платформі The RoAR.

**Редактор ROAR** – кращий в своєму класі інструмент для створення власного високоякісного об'єкту доповненої реальності всього за пару кліків. Дана платформа підходить для створення доповненої реальності для авторів, бізнесу і підприємств. Вона проста та легка у використанні та не потребує особливих технічних навичок та знання якоїсь мови програмування для створення своєї продукції. Редактор ROAR містить всі основні функції доповненої реальності, включаючи відео, аудіо, кнопки дій, зображення, текст, 3D-моделі тощо. Дана платформа дозволяє працювати з 3 варіантами створення елементів доповненої реальності – на основі зображення-маркеру, Web-AR та ARKit & ARCore. Приклади проектів реалізованих з використанням даного редактору можна переглянути за посиланням <https://theroar.io/roar-gallery/>.

## 2. Створення AR-елементу на The RoAR.

Для створення власного AR-об'єкту спочатку треба зареєструватись на сайті <https://theroar.io/>.

Наступним кроком необхідно обрати вкладку «**Create AR**». Далі необхідно обрати режим «**Target image**».

Далі необхідно обрати зображення, що буде маркером у даному випадку. Це може бути будь-яке зображення розміром мінімально 300x300 пікселів, наприклад:

- частина зображення - логотип, реклама в журналі, етикетка, частина білборда або сітілайта;
- повне зображення - сторінка журналу або газети, СРГ (їжа та напої), листівка або повне зображення рекламного щита.
- Крім того, маркер також може бути:
  - зображення на сайті;
  - статичне зображення або логотип під час телереклами;
  - цифрові каталоги, наприклад .pdf або .ppt;

- статичне зображення на DOOH.

Є також можливість використання шаблонів редактора ROAR, щоб підібрати коректний маркер для створюваного елемента доповненої реальності. Після проходження перевірки редактором на коректність даного зображення та відсутність схожості з будь-яким іншим існуючим маркером необхідно наповнити елемент цифровим контентом.

### 3. Наповнення елемента доповненої реальності

Інтерфейс редактора має наступний вигляд, представлений на рисунку 3.1.1.

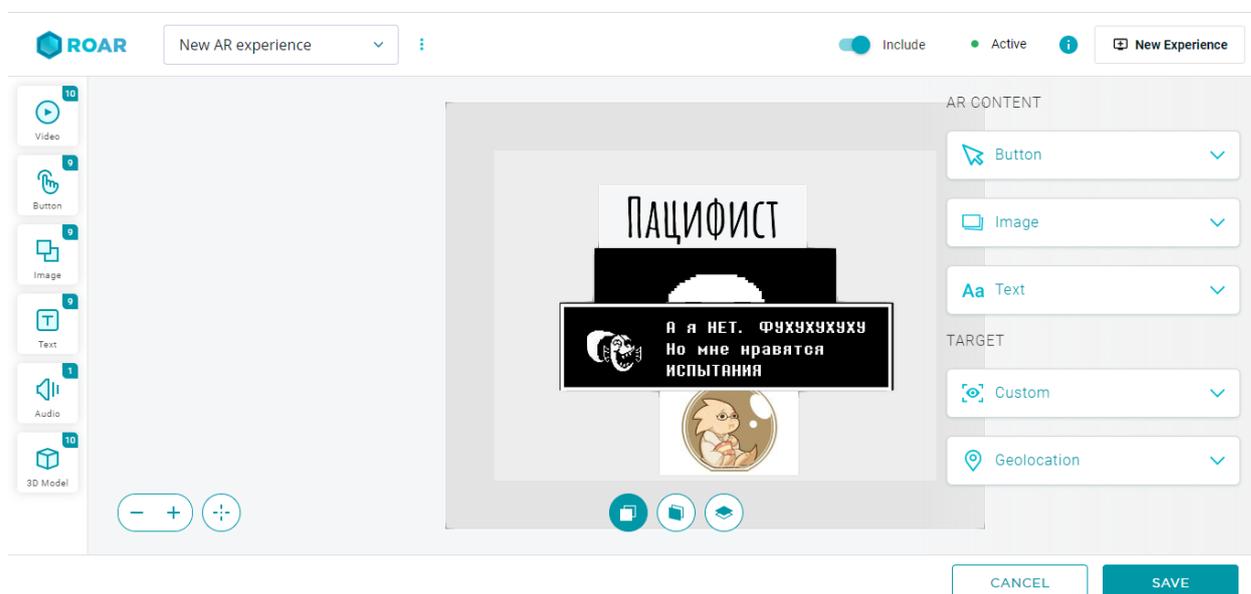


Рисунок 3.1.1 – Інтерфейс редактора

Як можна побачити, робочу частину редактора можна поділити на 3 частини: зона можливих елементів, якими можна наповнити контент AR-об'єкту (ліва частина); безпосередньо сама зона редагування та представлення самого елемента (центральна частина) з можливістю побачити як все виглядатиме під різним кутом зору та зона редагування кожного елемента контенту (права частина). Елемент може містити текст, зображення, аудіо- та відеоінформацію, кнопку та 3D-об'єкт.

Редактор ROAR дозволяє завантажити .zip-архів з 3D-елементами та налаштувати додаткові параметри, такі як ширина, висота, обертання тощо.

3D-вміст має бути підготовлений у належному форматі. Платформа ROAR підтримує такі формати:

- **\*.obj** – для 3D -моделі
- **\*.png, \*.jpg, \*.tga** – для текстур
- **\*.mtl** – з усіма наборами

Редактор також спрощує використання відеоконтенту AR. Відеофайл можна завантажити із користувацького жорсткого диска або скопіювати / вставити посилання YouTube \*. В результаті AR-відео транслюватиметься з хмари і відтворюватися безпосередньо на цільовому зображенні. Крім того, можна використовувати хромакей для створення високоякісного відео для AR-контенту.

**Кнопки із закликком до дії** на основі доповненої реальності надають можливість переспрямувати користувача на будь-який веб -сайт або, наприклад, завантажити мобільний додаток або для інших дій в залежності від концепції майбутнього продукту, наприклад, покупки товару. Можна використовувати шаблони кнопок ROAR або завантажувати власні кнопки.

Редактор дозволяє редагувати кегль, накреслення, гарнітуру, вирівнювання шрифту, його розміщення стосовно інших елементів, кут повороту та колір, а також робити його інтерактивним, перетворюючи його у кнопку переходу за допомогою посилання. Для впровадження того чи іншого елемента у контент доповненої реальності треба затиснути цей елемент лівою клавішею миші та перетягнути його до області редагування. Після цього можна прикріпити файл відповідного формату до цього елемента та виконати його редагування.

#### **4. Тестування та публікація елемента**

Після проведення усіх необхідних дій з розроблюваним елементом доповненої реальності необхідно опублікувати цей елемент для подальшої перевірки на коректність відтворювання. Для цього слід натиснути кнопку «**Save**», після чого з'явиться діалогове вікно з можливістю переходу до

**PlayMarket** для **Android**-систем або **App Store** для **Apple**-систем, а також QR-код, для завантаження та встановлення **ROAR Augmented Reality App** – спеціального додатку для відтворення елементів доповненої реальності, який необхідно завантажити на свій телефон чи планшет. У даному вікні слід натиснути кнопку «**Publish**». Після цього відбувається автоматичний перехід до вкладки **My RoAR**, де у вкладці **Published** знаходиться опублікований елемент. Далі треба навести курсор на даний елемент і натиснути на зображення ока у правому верхньому куті (при наведенні на нього висвічується назва **Preview**). Після цього відкривається автоматично наступне вікно, у якому прописаний наступний порядок дій для тесту елемента:

1. Завантажити додаток по QR-коду (якщо цього не було зроблено раніше).
2. Запустити додаток ROAR та увійти, використовуючи облікові дані.
3. Відкрити повноекранну версію елемента.
4. На телефоні обрати другу вкладку із зображенням кола та зісканувати зображення на екані.

## **Хід роботи та зміст завдання**

### **1. Ідея проєкту/розробка концепції**

Описати ідею продукту, його цільове призначення, аудиторію, галузь застосування (AR для друкованої продукції, геолокація та туризм, освіта, маркетинг і реклама тощо).

### **2. Мета створення проєкту**

Розробити прототип мобільного застосунку (Android OS) на базі технології доповненої реальності, що матиме освітню або розважальну, або соціальну тощо цінність (вказати яку саме для вашої ідеї), доповнивши віртуальним контентом продукт (вказати який саме, назва, вигляд, конструкція продукту, що матиме елементи доповненої реальності).

### **3. Предпроектна підготовка/Формат та структура додатку**

- вказати технологію доповненої реальності, що використовуватиметься при реалізації проєкту (маркерна, безмаркерна);

- запроєктувати необхідну кількість сторінок/екранів/тематичних розділів тощо, створити карту проєкту. Відобразити ієрархічну структуру та варіанти переходів між сторінками, якщо такі передбачено ідеєю.

- створити/підібрати та описати маркери доповненої реальності. Роздрукувати при необхідності та перевірити їх працездатність.

- підібрати/розробити 2D та 3D матеріали до додатку (відео, зображення, аудіо, текст, інфографіка, зображення тощо), що відповідатимуть ідеї та нестимуть певний інформаційний посил.

**4. Реалізація проєкту з використанням онлайн сервісу (за допомогою редактору RoAR).** Використати якомога більше типів цифрового контенту.

**5. Запуск та описати особливості підтримки проєкту.**

**6. Створити звіт з виконаної роботи з супровідними скріншотами.**

### **Контрольні запитання**

1. Доповнена реальність. Її особливості застосування.
2. Технології доповненої реальності. Їх характеристика.
3. Програмне забезпечення для створення AR-елементів.

### **Перелік літератури за темою**

1. Яковлев Б. С. Классификация и перспективные направления использования технологии дополненной реальности [Текст] / Б. С. Яковлев, С. И. Пустов // Известия Тульского государственного университета. Технические науки. – 2013.

2. Технологии и алгоритмы для создания дополненной реальности [Електронний ресурс] – Режим доступу: <https://www.mais-journal.ru/jour/article/view/211/221>, вільний. – Назва з екрану – Мова рос.

3. Технология виртуальной и дополненной реальности [Электронный ресурс] – Режим доступа: <https://cyberleninka.ru/article/n/tehnologii-virtualnoy-i-dopolnennoy-realnosti-vozmozhnosti-i-prepyatstviya-primeneniya>, вільний. – Назва з екрану – Мова рос.

4. Технологии и алгоритмы для создания дополненной реальности [Электронный ресурс] – Режим доступа: <https://www.mais-journal.ru/jour/article/view/211/221>, вільний. – Назва з екрану – Мова рос.

5. Технологии дополненной реальности [Электронный ресурс] – Режим доступа: <http://dgng.pstu.ru/conf2017/papers/110/>, вільний. – Назва з екрану – Мова рос.

6. Дополненная реальность [Электронный ресурс] – Режим доступа: <http://old.tusur.ru/filearchive/reports-magazine/2010-1-2/161-165.pdf/>, вільний. – Назва з екрану – Мова рос.

## КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 3.2

### СТВОРЕННЯ ЕЛЕМЕНТІВ ДОПОВНЕНОЇ РЕАЛЬНОСТІ ЗА ДОПОМОГОЮ UNITY

**Мета:** Створення AR-елементів за допомогою програмного забезпечення Unity.

#### Теоретичні відомості

##### 1. Загальні відомості про ресурс

**Unity** – багатоплатформовий інструмент для розробки відеоігор і застосунків, і рушій, на якому вони працюють. Створені за допомогою Unity програми працюють на настільних комп'ютерних системах, мобільних пристроях та гральних консолях у дво- та тривимірній графіці, та на пристроях віртуальної чи доповненої реальності. Застосунки, створені за допомогою Unity, підтримують DirectX та OpenGL.

Програма-редактор Unity працює на Windows, macOS і Linux, а сам рушій може запускатися на 25 платформах, а саме iOS, Android, Tizen, Windows, Mac, Linux, PlayStation Xbox One, Apple ARKit та інші. Ігрова логіка пишеться за допомогою мови C#, раніше також була можливість використовувати Boo та JavaScript, але розробники відмовились від їх підтримки.

##### 2. Функціональні можливості

Редактор Unity має інтерфейс, що складається з різних вікон, які можна розташувати на свій розсуд. Завдяки цьому можна проводити налагодження розроблюваного об'єкту прямо в редакторі. **Головні вікна** – це оглядач ресурсів проєкту **Project**, інспектор поточного об'єкта **Inspector**, вікно попереднього перегляду **Game**, оглядач сцени **Scene** та оглядач ієрархії ресурсів **Hierarchy**.

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, може бути тег (мітка) і шар, на якому він повинен відображатися. Різні моделі можуть об'єднуватися в набори (ассети) для швидкого доступу до них. 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. 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. Фон сцени, видимий через камеру, типово зображає небо, але може презентувати й інше довкілля. Unity підтримує файли 3ds Max, Maya, Softimage, Blender, modo, ZBrush, Cinema 4D, Cheetah3D, Adobe Photoshop, Adobe Fireworks та Allegorithmic Substance.

### 3. Vuforia

**Vuforia** – це платформа доповненої реальності й інструментарій розробника програмного забезпечення доповненої реальності (Software Development Kit – SDK) для мобільних пристроїв, розроблені компанією Qualcomm. Vuforia використовує технології комп'ютерного зору, а також відстеження плоских зображень і простих об'ємних реальних об'єктів (наприклад, кубічних) в реальному часі. З версії 2.5 Vuforia розпізнає текст, а з 2.6 – має можливість розпізнавати циліндричні маркери.

Можливість візуалізації зображень дозволяє розробникам розташовувати і орієнтувати віртуальні об'єкти, такі, як 3D-моделі і медіаконтент, в поєднанні з реальною сценою при відображенні її на дисплеях мобільних пристроїв. Віртуальний об'єкт позиціонується на реальному образі так, щоб точка зору спостерігача співвідносилася з ними однаковим чином для досягнення відчуття, що віртуальний об'єкт є частиною реального світу. Vuforia підтримує різні типи маркерів, в тому числі безмаркерні технології Image Target, тривимірні багатоцільові мішені Multi-Target та інші. Додаткові можливості дозволяють уникнути ефекту маскування об'єктів з використанням так званих «Віртуальних кнопок» («Virtual Buttons»), забезпечити селекцію об'єктів і можливість програмно створювати і реконфігурувати їх у рамках коду, що самостійно модифікується. Vuforia надає інтерфейси програмування додатків на мовах C++, Java, Objective-C, і .Net через інтеграцію з Unity. Додатки доповненої реальності, створені на платформі Vuforia, сумісні з широким спектром пристроїв, включаючи iPhone, iPad, смартфони та планшети на Android з версії 2.2 і процесором, починаючи з архітектур ARMv6 або 7 з можливістю проведення обчислень з плаваючою комою.

### Інструкція з виконання

1. Зареєструватись на офіційному сайті Unity <https://unity.com/ru> та завантажити програмне забезпечення **Unity Hub** з [https://store.unity.com/download-nuo?\\_ga=2.163898495.1926477259.1629005417-2109006814.1629005416&\\_gl=1%2Amjhkbe%2A\\_ga%2AMjEwOTAwNjgxNC4xNjI5MDA1NDE2%2A\\_ga\\_1S78EFL1W5%2AMTYyOTAwNTQxNS4xLjEuMTYuOTAwNTg1NC42MA](https://store.unity.com/download-nuo?_ga=2.163898495.1926477259.1629005417-2109006814.1629005416&_gl=1%2Amjhkbe%2A_ga%2AMjEwOTAwNjgxNC4xNjI5MDA1NDE2%2A_ga_1S78EFL1W5%2AMTYyOTAwNTQxNS4xLjEuMTYuOTAwNTg1NC42MA).
2. Інсталювати завантажений файл за відповідними вказівками.

3. Завантажити з <https://developer.android.com/studio> застосунок **Android Studio** останньої версії та інсталиувати його. Слід врахувати декілька особливостей:

- Після встановлення необхідно завантажити програму де, слідує вказівкам, необхідно встановити додаткові компоненти;
- У меню зі встановленням компонентів у підпункті із вказуванням шляху для встановлення Android Studio SDK необхідно забезпечити англійський шлях до папки зберігання цих компонентів Android\SDK (у випадку наявності українсько- чи російськомовних назв буде виникати зауваження *Your SDK location contains non-ASCII characters*);
- Для подолання попередньо описаної проблеми можна локальному диску C створити папку Android, а у ній папку SDK, і потім вказати шлях до цієї папки;
- Після всіх налаштувань необхідно дочекатися встановлення усіх компонентів.

4. Завантажити **додаткові налаштування** за посиланням [https://dl.google.com/android/repository/tools\\_r25.2.3-windows.zip](https://dl.google.com/android/repository/tools_r25.2.3-windows.zip) та розархівувати файли у порожню нову папку, яку назвати **tools** та після виконання попереднього пункту необхідно замінити цією папкою папку з аналогічною назвою у створеній у попередньому пункті папці SDK.

5. Завантажити встановлений **Unity Hub**, обрати шлях для встановлення програмного забезпечення, підтвердити свій попередньо зареєстрований профіль у підменю Create Unity ID, після цього обрати режим Empty 3D у підменю Select Microgame перейти до підменю Launch Unity, дочекатися встановлення Unity та завантажити його.

6. Завантажити знову **Unity Hub** і перейти у меню **Instal**, де обрати параметри у інстальованому елементі **Unity** («:»), у якому обрати пункт **Add Modules**. У вікні, що з'явилося, обрати модуль **Android Build Support**, натиснути **Done** та дочекатися його встановлення.

7. Зареєструватися на офіційному сайті Vuforia <https://developer.vuforia.com/> та перейти у вкладку **Develop**, де спочатку

обрати меню **Licenses Manager**, що дозволить створити спеціальний **Development Key** для роботи з **Vuforia**. У цьому меню необхідно створити новий **Key** шляхом натискання на кнопку **Get Development Key**. У вкладці, що з'явилася необхідно дати назву та натиснути кнопку **Confirm**, погоджуючись з умовами. Після цього треба зайти у створений ключ та скопіювати ліцензію.

8. Наступним кроком необхідно створити маркер-зображення. Для цього спочатку необхідно скачати зображення, яке буде маркером. Далі треба перейти у меню **Target Manager** у вкладці **Develop** та натиснути кнопку **Add Database**. У меню, що з'явилося, необхідно дати назву маркеру, обрати тип **Device** та натиснути кнопку **Create**. Потім обрати створений маркер та у новій вкладці натиснути кнопку **Add Target**, у меню, що з'явилося, обрати тип **Single image**, обрати своє зображення у пункті **File**, встановити ширину **Width**, наприклад, **5** і натиснути кнопку **Add**. Після цього у характеристиці **Rating** зображення з'явиться певна кількість зірок – від 1 до 5, що відповідає рівню розпізнавання зображення. Найкращим варіантом є таке зображення, у якого рейтинг становить 5 зірок – воно легко та швидко розпізнається майже за будь-яких умов, найгіршим – 1. Краще підбирати зображення із рейтингом неменше 3 зірок. Наступним кроком необхідно клацнути на кнопку **Download Database (All)**, де обрати **Unity Editor** та натиснути **Download** та дочекатися завантаження, після чого відкрити завантажений файл та імпортувати створену базу до середовища **Unity**.

9. Завантажити з <https://developer.vuforia.com/> з меню **Downloads** файл пункту **Add Vuforia Engine to a Unity Project or upgrade to the latest version**, відкрити його та натиснути кнопку **Import** у меню, що з'явиться у середовищі **Unity** та дочекатися завершення операції.

10. Відкрити встановлену програму **Unity**, обрати вкладку **File**, обрати пункт **Build Settings**. У новому вікні обрати систему, під яку розроблятиметься продукт (у даному випадку **Android**) та натиснути кнопку **Switch**. У вкладці **Window** середовища **Unity** обрати пункт **Package Manager**, де знайти та встановити пакети **AR Foundation** та **AR Core XR**

**Plugin.** Також необхідно переформатувати обраний проект під систему Android. Для цього треба зайти в меню **File/Build Settings**. У новому вікні обрати **Android** та натиснути кнопку **Switch Platform**.

11. Наступним кроком необхідно створити AR-камеру. Для цього можна клацнути правою кнопкою миші у полі **Hierarchy** та вибрати вкладку **Vuforia Engine**, де обрати варіант **AR Camera** (або наступний шлях **GameObject / Vuforia Engine / AR Camera**).

12. Далі необхідно налаштувати пакет **Vuforia Engine**. Для цього необхідно обрати новостворену камеру у полі **Hierarchy**, потім перевести курсор у поле **Inspector** та обрати пункт **Open Vuforia Engine Configuration** та у новому меню, що відкрилося у полі вводу ліцензії ввести скопійований у 8 пункті ключ.

13. Далі необхідно зробити прив'язку до попередньо створеного маркера у пункті 8. Спочатку необхідно створити зображення-мітку в робочому полі **Scene**. Для цього слід клацнути правою кнопкою миші у полі **Hierarchy** та вибрати вкладку **Vuforia Engine**, де обрати варіант **Image Target**. Якщо зображення одразу не з'явилося, необхідно у полі **Inspector** створеного елемента **Image Target** обрати пункт **Image Target Behaviour**, потім у пункті **Type** обрати варіант **From Database**, після чого у пункті **Database** обрати назву імпортованої бази з маркером у пункті 9. Після виконання даних дій відповідний маркер-зображення з'явиться у полі **Scene** у початковій точці. Його у полі **Hierarchy** треба перетягнути створене зображення на камеру, аби присвоїти їй даний маркер.

14. Наступним кроком необхідно наповнити маркер. Для даної роботи необхідно буде створити логічну завершену сцену з анімованою 3D-моделлю, 3D-декораціями та музичним наповненням. Для завантаження анімованої 3D-моделі можна завантажити звичайну фігуру в форматі .fbx (бажано з витягнутими вбік руками, що нагадують літеру Т (див. рис. 3.2.1)), розархівувати завантажений архів (буде містити 2 папки – **Source**, де безпосередньо зберігається файл 3D-моделі, та **Texture**, де зберігаються текстури, що формують зовнішній вигляд моделі) наприклад, з

<https://sketchfab.com/>, а потім на сайті <https://www.mixamo.com> обрати необхідну анімацію і, натиснувши на кнопку **Upload Character** та виконавши усі вказівки, підставити свою 3D-модель та завантажити її. 3D-декорації можна також підвантажити та накласти на них текстуру або ж створити їх самостійно за допомогою можливостей Unity шляхом **Hierarchy/ПКМ/Create/3D-object** і потім обрати необхідний об'єкт та налаштувати його за допомогою поля **Inspector**.

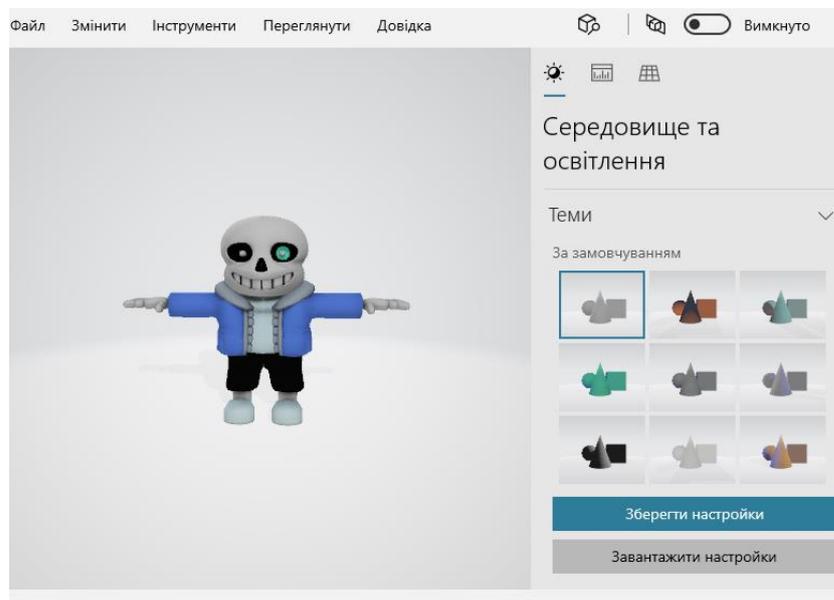


Рисунок 3.2.1 – Приклад 3D-моделі

15. Для легшого пошуку елементів та структування поля **Project** можна створювати папки. Дане поле поділене на дві частини – менша частина показує усі папки та елементи проекту, права – склад папок. Усі основні елементи проекту містяться у папці **Assets**. Для створення папки або будь-якого іншого елемента треба клацнути правою кнопкою миші у правій частині поля **Project**, вибрати пункт **Create** та обрати **Folder** або будь-який інший необхідний елемент. Далі треба назвати дану папку для легкого пошуку елементів, зайти в неї та перетягнути в це поле завантажену анімовану модель та текстуру до неї, яка зберігається в папці **Texture**. Далі необхідно витягнути цю фігуру на робочу область поля **Scene**, обрати її у полі **Hierarchy**, відредагувати її розмір та положення у полі **Inspector**, а

також підв'язати її під AR-маркер шляхом перетягування елемента з назвою моделі на назву маркеру в меню **Hierarchy**. Далі у даному полі треба створити новий матеріал (ПКМ/Create/Material), обрати його, перетягнути обрану текстуру у поле **Inspector** даного матеріалу у пункт **Albedo**, а потім обрати необхідний елемент, на який необхідно накласти текстуру у полі **Hierarchy**, та перетягнути створений матеріал у поле **Materials/Elements**. Таким чином обрана текстура накладеться на модель.

16. Далі необхідно налаштувати анімацію. Для цього необхідно обрати 3D-модель у поля **Project**, клацнути на ній ПКМ та обрати **Create/Animator Controller**. Далі обрати створений елемент та завантажити його. У новому полі клацнути ПКМ та створити новий стан **Empty**. Новоствореному стану дати назву, обрати його, в пункт Motion поля **Inspector** перетягнути анімацію моделі з поля **Project**, зациклити анімацію (завантажити створений стан та поставити галку у пункті **Loop**). Далі необхідно обрати 3D-модель у полі **Hierarchy**, у полі **Inspector** натиснути кнопку **Add Component** та обрати **Animator**, перетягнути створений **Animator Controller** у поле **Controller** створеного компонента. Таким чином 3D-модель буде рухатися.

17. Для додавання музичного супроводу необхідно у папці **Assets** створити нову папку та перетягнути до неї будь-яку музику, що підходить під сюжет. Потім у полі **Hierarchy** створити новий порожній об'єкт (**Create Empty**), назвати його **Music**, обрати його та у полі **Inspector** натиснути кнопку **Add Component**, де додати компонент **Audio Source**, у поле **Audio Clip** якого перетягнути завантажену музику з поля **Project**. Музику також можна зациклити шляхом натискання галки у полі **Loop** компоненту **Audio Source**. Створений елемент **Music** у полі **Hierarchy** треба перетягнути у маркер у тому ж полі, аби музика підв'язалася під маркер.

18. Після створення усіх необхідних компонентів необхідно створити додаток. Для цього треба перейти **File/Build Settings** та зробити певні налаштування – натиснути кнопку **Player Settings**, обрати у новому вікні **Settings for Android**, у ньому вкладку **Other Settings**, та поставити

галку **Auto Graphics API** та у полі **Minimum API Level** версію **Android 7.0**. Це дозволить уникнути помилок при створенні додатку. Далі треба повернутись до меню **Build Settings** та натиснути кнопку **Build**. При виборі папки для збереження елемента необхідно слідкувати, аби в шляху знаходження файлу були лише латинські літери.

19. Для тестування додатку необхідно завантажити створений APK на телефон, встановити його, завантажити та навести на маркер на екрані або ж надрукований маркер.

### Хід роботи

1. Встановити Unity та всі необхідні компоненти.
2. Створити 3D-модель фону.
3. Створити анімовану 3D-модель, що підходить під сюжет.
4. Створити музичний супровід.
5. Елемент доповненої реальності повинен мати якусь логіку та ідею.
6. Створити звіт з виконаної роботи з супровідними скріншотами.

### Контрольні питання

1. Unity. Операційні системи, під які можна розробляти додатки та мови програмування, які можна застосовувати.
2. Функціональні можливості Unity.
3. Vuforia та її особливості.

## КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 3.3

### СТВОРЕННЯ ЕФЕКТІВ ЗА ДОПОМОГОЮ UNITY ТА ДОДАВАННЯ ЇХ ДО ЕЛЕМЕНТІВ ДОПОВНЕНОЇ РЕАЛЬНОСТІ

**Мета:** Створення ефектів за допомогою програмного забезпечення Unity та впровадження їх у середовище елемента доповненої реальності.

#### Теоретичні відомості

Одним із застосовуваних ефектів Unity є **Particle System**, на базі якого будується різноманітна анімація в додатках – сніг, вогонь, феєрверк та інші. Компонент Particle System імітує рідкі субстанції на зразок різних рідин, хмар і чогось пов'язаного з вогнем шляхом генерації й анімації в сцені великої кількості невеликих 2D-елементів. Даний компонент має безліч налаштувань, що поділені на модулі. Біля заголовку кожного модуля розташовані чекбокси, які можна відзначати для включення або відключення тих чи інших налаштувань.

Крім модулів, в інспекторі даного компоненту є ще кілька інших елементів управління. Кнопка **Open Editor** відображає опції в окремому вікні редактора, яке також дозволяє редагувати одночасно кілька систем. А чекбокс **Resimulate** відповідає за те, будуть чи не будуть застосовуватися зміни властивостей до щойно створених системою часток у цей же момент. Кнопка **Wireframe** відображає контури меш-об'єктів, щоб показати частки, що знаходяться за ними.

Існують такі модулі:

- **Particle System Module** – містить глобальні налаштування, які впливають на всю систему в цілому;
- **Emission Module** – властивості цього модуля впливають на частоту й синхронізацію генерації частинок;

- **Shape Module** – використовується для зміни форми Particle System, який генерується, і місця, звідки генеруються самі частинки;
- **Velocity Over Lifetime Module** – для зміни швидкості частинок протягом їх життєвого циклу;
- **Limit Velocity Over Lifetime Module** – управляє тим, як буде зменшуватися швидкість часток протягом їхнього життєвого циклу;
- **Force Over Lifetime** – можна призначити прискорення частинок за рахунок впливу зовнішніх сил;
- **Color Over Lifetime** – визначає як частки з плином часу змінюють колір і прозорість;
- **Color By Speed Module** – для зміни кольору частинок відповідно до їх швидкості за секунду;
- **Size Over Lifetime** – для зміни розміру частинок відповідно до кривої, яку можна налаштувати в цьому модулі;
- **Size by Speed** – для зміни розміру частинок відповідно їх швидкості в одиницю відстані в секунду;
- **Rotation Over Lifetime Module** – для організації обертання частинок в потрібному порядку під час їх руху;
- **Rotation by Speed** – для зміни обертання частинок відповідно їх швидкості в одиницю відстані в секунду;
- **External Forces Module** – модифікує ефект wind zones на частинках, що генеруються Particle System;
- **Sub Emitter** – дозволяє налаштовувати sub-emitters;
- **Texture Sheet Animation Module** – дозволяє використовувати свою текстуру як сітку невеликих зображень, які можна перемикати від кадру до кадру у вигляді анімації;
- **Rendering** – визначає як трансформується текстура частинок або меш.

У наступній таблиці 3.3.1 представлені параметри **Particle System**, та характеристики, які вони регулюють.

Таблиця 3.3.1 – Параметри Particle System

Назва параметру	Опис характеристики, що регулюється
<b>Particle System Module</b>	
Duration	Тривалість часу, протягом якого буде працювати система
Looped	Якщо підключена, то система почне свою роботу в місці завершення свого duration циклу і почне все заново.
Prewarm	Якщо підключена, то система буде ініціалізована так, ніби вона вже пройшла через весь цикл своєї роботи (працює виключно тоді, коли Looping також включено)
Start Delay	Затримка в секундах до старту системи генерації частинок після її включення
Start Lifetime	Початкова тривалість життя частинок
Start Speed	Початкова швидкість кожної частинки у відповідному їй напрямку
Start Size	Початковий розмір кожної частинки
Start Rotation	Початковий кут повороту кожної частинки
Start Color	Початковий колір кожної частинки
Gravity Multiplier	Зміна значення гравітації. Значення 0 призведе до відключення гравітації.
Inherit Velocity	Характеризує наскільки самі частинки рухаються швидше чи повільніше за систему в цілому
Simulation Space	Визначає систему відліку для частинок
Play on Awake	Визначає автоматичність процесу роботи системи
Max Particles	Максимальна кількість частинок, що одночасно знаходяться у системі
<b>Emission Module</b>	
Rate	Кількість частинок, що генеруються за одиницю часу або пройденого шляху
Bursts	Можливість додаткових частинок генеруватися в потрібний момент часу
<b>Shape Module</b>	
Shape	Форма Particle System
Random Direction	При підключенні створює довільний початковий напрямок частинок
Radius	Радіус Particle System (тільки для сфери, півсфери і конуса)
Angle	Кут конуса Particle System (тільки для конуса)
Box X, Y, Z	Ширина, висота і глибина кубічної форми (тільки для кубів)
Mesh	Меш, що задає форму генератора (тільки для мешів)
Emit from Shell	Визначає, з якої сторони Particle System будуть генеруватися частинки – ззовні чи зсередини (тільки для сфер і півсфер).
Emit from	Визначає частину конуса, з якої буде відбуватися генерація частинок: (тільки для конусів).
<b>Velocity Over Lifetime Module</b>	
X, Y, Z	Швидкість по осях X, Y і Z.
Space	Визначає систему відліку для осей X,Y,Z
<b>Limit Velocity Over Lifetime Module</b>	
Speed	Ліміт швидкості
Space	Визначає відношення обмежень швидкості до локального або світового простору
Dampen	Характеризує значення зниження швидкості частинок в момент досягнення ними максимальної швидкості

## Продовження таблиці 3.3.1

Назва параметру	Опис характеристики, що регулюється
<b>Force Over Lifetime</b>	
X, Y, Z	Сила, що застосовується до частинок по осям X, Y та Z
Space	Визначає простір застосування сили
<b>Rotation by Speed</b>	
Angular Velocity	Швидкість обертання в градусах в секунду
Speed Range	Нижня та верхня межі діапазону швидкості, на яку накладається крива розміру
<b>Collision</b>	
World/Planes popup	Визначення режиму World чи Planes
Planes	Розгортається список перетворень, який визначає площину зіткнень (тільки для режиму Planes)
Visualization	Визначення відображення Гизмо collision plane у видовому вікні сцени (тільки в режимі Planes)
Scale Plane	Розмір площини, що використовується для візуалізації (тільки в режимі Planes)
Dampen	Частина швидкості, що втрачається частинкою після зіткнення
Bounce	Частина швидкості частинки, переданої поверхні, від якої вони відбиваються
Lifetime Loss	Частина від загального часу життя частинки, яка втрачається при зіткненні.
Min Kill Speed	Мінімальна швидкість, нижче якої відбуватиметься видалення частинок після зіткнення
Particle Radius	Приблизний розмір часток, що використовується для уникнення відсікання з площинами зіткнення (тільки в режимі Planes).
Collides With	Частинки будуть стикатися тільки з об'єктами, розташованими на виділених шарах (тільки в режимі World).
Collision Quality	Визначає непроникність зіткнення (тільки в режимі World).
Send Collision Messages	Якщо підключено, то зіткнення частинок можуть бути зафіксовані через функцію скрипта OnParticleCollision.
<b>Texture Sheet Animation Module</b>	
Tiles	Кількість секцій, на які текстура буде розбита по осям X і Y
Animation	Режим анімації
Random Row	Навмання вибирається ряд зображень для відтворення анімації
Row	Вибирається конкретний ряд з сітки зображень для відтворення анімації
Frame Over Time	Крива, яка визначає те, як з плином часу змінюються кадри анімації
Cycles	Кількість повторень анімації протягом періоду життя частинок
<b>Rendering</b>	
Render Mode	Режими рендерингу
Camera Scale	Величина розтягування, що застосована пропорційно до руху камери (тільки в режимі Stretched Billboard).
Speed Scale	Величина розтягування, що застосована пропорційно до швидкості частинок (тільки в режимі Stretched Billboard).

## Кінець таблиці 3.3.1

Назва параметру	Опис характеристики, що регулюється
Length Scale	Величина розтягування, що застосована пропорційно до довжині частинок (тільки в режимі Stretched Billboard).
Mesh	Кількість мешів (тільки для режиму Mesh).
Material	Матеріал, що формує текстуру частинок.
Sort Mode	Порядок розташування системи
Sorting Fudge	Зсув порядку сортування частинок
Cast Shadows	Наявність тіні від частинок
Receive Shadows	Визначає, чи повинні тіні відкидатися на частинки
Max Particle Size	Найбільший розмір частинок по відношенню до розміру вікна проекції.

### Приклади ефектів та етапи їх створення

#### 1. Ефект снігу, що падає

Даний ефект створюється наступним чином:

1. У полі **Hierarchy** створюється елемент **Particle System** шляхом ПКМ/Effect/Particle System.

2. У меню **Inspector** цього елемента у полі **Shape/Shape** обирається варіант **Box**.

3. У полі **Transform** задається розмір та положення площі, яку займатиме елемент (розмір можна проставити 10,10, 1 (поле Z не чіпати)).

4. Далі необхідно провести попереднє налаштування самого елемента. Для цього проставити наступні параметри:

- **Prewarm** – поставити галку;
- **Start Lifetime** – 10;
- **Start Speed** – 0 (аби частинки не рухалися вгору);
- **Start Size** – встановити Random Between Two Constants і проставити невеликі значення (наприклад, від 0,01 до 0,03);
- **Velocity over Lifetime** – підключити та поставити значення X (від -2 до 2), Y (від -2 до 2), Z (від -1 до -2) – для того, щоб сніг йшов вниз та нерівномірно, а з певним відхиленням від прямого напрямку;

- **Color over lifetime** – обрати цей пункт, клацнути на білому полі **Color**, встановити мітки 10% та 90% на верхній частині діапазону та встановити значення Alpha 0 для 0% та 100%.

5. Далі необхідно налаштувати зовнішній вигляд снігу. Для цього спочатку створити новий документ у Photoshop розміром 100x100 пікселів у форматі RGB 72dpi. Далі залити фон чорним, розділити навпіл по горизонталі та вертикалі за допомогою направляючих та створити новий шар. На цьому шарі створити 4 елементи (кожний у своїй четверті) за допомогою пензлика із розміром Opacity 35% та розміром 6 пікселів. Має вийти щось схоже на приклад, наведений на рисунку 3. Зберегти його у форматі .jpeg та втягнути цей файл у поле **Project** у новостворену папку у папці **Asset**, вибрати його та поставити режим у полі **Texture Type** «**Sprite 2D and UI**», натиснути **Apply**;

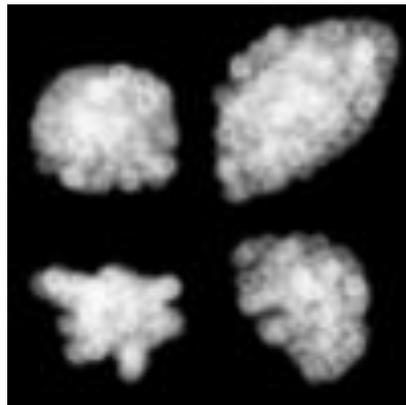


Рисунок 3.3.1 – Приклад текстури

6. Створити у цій же папці **Material**, поставити у полі **Inspector** у полі **Shader** обрати **Particles/Standard Unlit** та у полі **Render Mode** режим **Additive**, перетягнути створене у попередньому пункті зображення на поле **Albedo** та перетягнути даний матеріал у поле **Renderer/Material** поля **Inspector** елементу **Particle System**, створеного у попередніх пунктах;

7. Далі треба налаштувати зовнішній вигляд сніжинок, аби це не був просто квадрат з 4 елементами. Для цього треба налаштувати:

- **Texture Sheet Animation** – підключити і встановити значення **Tiles X=2, Y=2**;

- **Frame over Time** – від 0 до 3,9996 (таким чином обирається наближений зовнішній вигляд сніжинки відповідно до створеного у 5 пункті елементу);

- **Emission** – обрати даний пункт та поставити значення **Rate Over Time**  $\pm 70$ .

Загалом можна ще позмінювати попередні параметри. У результаті буде отриманий ефект снігу, який можна підв'язати до маркера шляхом перетягування **Particle System** на маркер у полі **Hierarchy**

## 2. Ефект світлячків

Цей ефект чимось схожий на попередній. Етапи створення:

1. У полі **Hierarchy** створюється елемент **Particle System** шляхом ПКМ/Effect/Particle System.

2. У меню **Inspector** цього елементу у полі **Shape/Shape** обирається варіант **Sphere** та радіус 2.

3. У полі **Transform** задається розмір та положення площі, яку займатиме елемент.

4. Далі необхідно провести попереднє налаштування самого елементу.

Для цього проставити наступні параметри:

- **Start Speed** – -5 (аби частинки рухалися до центру);

- **Start Size** – встановити **Random Between Two Constants** і проставити невеликі значення (наприклад, від 0,1 до 0,15);

- **Limit Velocity over Lifetime** – підключити та поставити значення **Speed 0.5 Dampen 0.2** для того, щоб частинки рухалися більш плавно;

- **Size over lifetime** – обрати цей пункт, клацнути на полі **Size** та сформувати криву, схожу на представлену на рисунку 3.3.2. Це дозволить зробити зміну розміру більш плавною та реалістичною із певним затуханням на кінці;

- **Noise** – обрати даний пункт, встановити значення параметрів **Strength 1, Frequency 2** та **Scroll Speed 0,5**. Це дозволить зробити рух світлячків більш хаотичним.

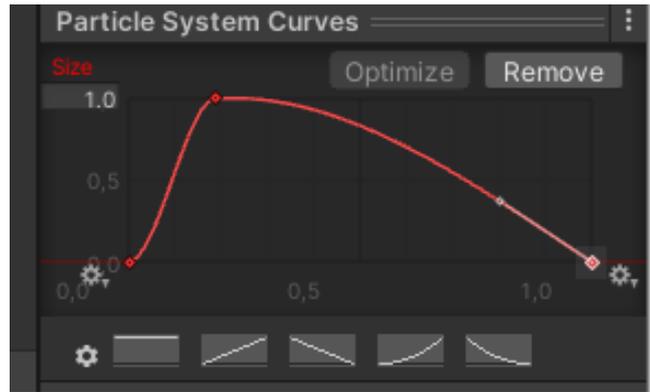


Рисунок 3.3.2 – Крива зміни розміру

5. Для створення шляху світляків необхідно обрати поле **Trail**, створити новий **Material** у новоствореній папці у папці **Assets**, у полі **Inspector** у полі **Shader** обрати **Particles/Standard Units** та у полі **Render Mode** режим **Additive** та перетягнути даний матеріал у поле **Renderer/Trail Materials** поля **Inspector** створеного **Particle System**.

6. Для налаштування даного сліду треба у полі **Trails** налаштувати наступні параметри – **LifeTime** 0.05, **Minimum Vertex Distant** поставити 0 та **Width Over LifeTime** обрати **Curve** та налаштувати криву, схожу на рисунок 4, та обрати значення початку відліку в 0,3 у верхньому лівому куті поля для редагування кривої.

7. Також треба налаштувати колір сліду. Для цього у полі **Start Color** треба обрати будь-який бажаний колір.

8. Наступним кроком необхідно створити самого світлячка. Для цього спочатку треба створити новий документ у Photoshop розміром 256x128 пікселів у форматі RGB 72dpi. Далі залити фон чорним, розділити навпіл по горизонталі та на 4 рівні частини по вертикалі за допомогою направляючих, а також розставити направляючі по межах області. В одній половині створити світлячка за допомогою еліпса, схожого на приклад на рисунку 3.3.3 зліва – спочатку голову, потім витягнуте тіло, копіюючи попередній шар, потім крила, роблячи копію з шару з тілом. Далі зробити групу з шарів з крилами, надати цій групі значення **Opacity** 50% та додати ефект тіні, де виставити

параметри, наведені на рисунку 6а. Далі скопіювати шар з головою та створити у кінці тільця трохи витягнуту частину, яка буде світити. Для цього шару додати ефект зовнішнього світіння та задати наступні параметри, наведені на рисунку 6б. Далі створити групу з попередніх шарів, створити її дублікат, перетягнути її в праву половину та трохи видозмінити положення крил відповідно до прикладу, представленою на рисунку 3.3.3. Потім зберегти файл у форматі .jpeg та втягнути цей файл у поле **Project** у новостворену папку у папці **Asset**, вибрати його та поставити режим у полі **Texture Type** «**Sprite 2D and UI**», натиснути **Apply**;

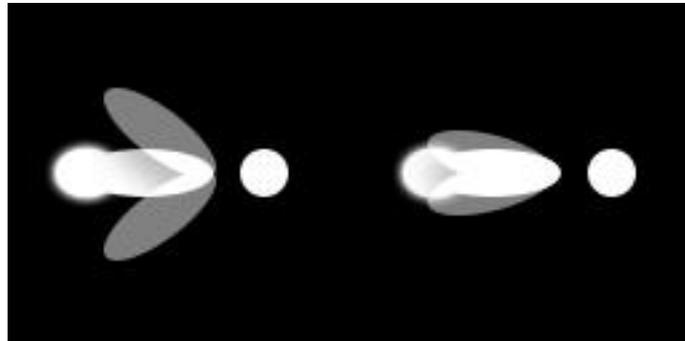
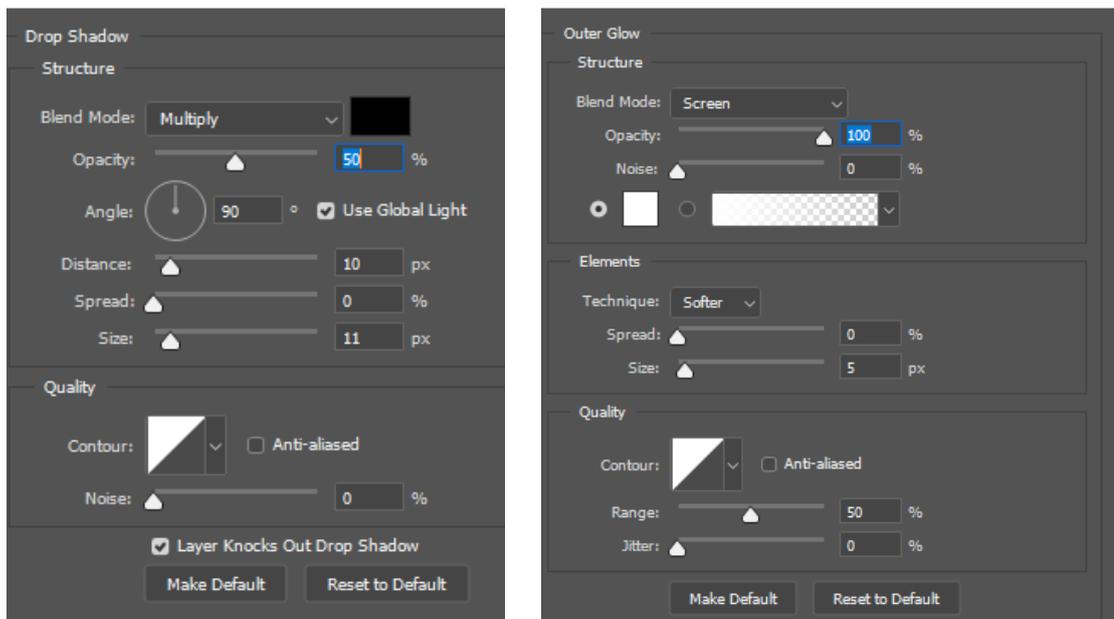


Рисунок 3.3.3 – Світлячок

9. Створити у цій же папці **Material**, поставити у полі **Inspector** у полі **Shader** обрати **Particles/Standard Unlit** та у полі **Render Mode** режим **Additive**, перетягнути створене у попередньому пункті зображення на поле **Albedo** та перетягнути даний матеріал у поле **Renderer/Material** поля **Inspector** елементу **Particle System**, створеного у попередніх пунктах;

10. Далі треба налаштувати зовнішній вигляд світлячків, аби це не був просто квадрат з 2 елементами. Для цього треба налаштувати:

- **Texture Sheet Animation** – підключити і встановити значення **Tiles X=2, Y=1** та **Cycles – 30**;
- **Renderer/Renderer Mode** – **Stretched Billboard**
- **Renderer/Length Scale** – встановити значення **-1** (таким чином світлячки трохи крутяться при русі);



а)

б)

Рисунок 3.3.4 – Параметри налаштування ефектів

Загалом можна ще позмінювати попередні параметри. У результаті буде отриманий ефект світлячків з анімацією крил, який можна підв'язати до маркера шляхом перетягування **Particle System** на маркер у полі **Hierarchy**

### 3. Ефект вогню

Даний ефект поєднує декілька елементів **Particle System**. Для його створення треба:

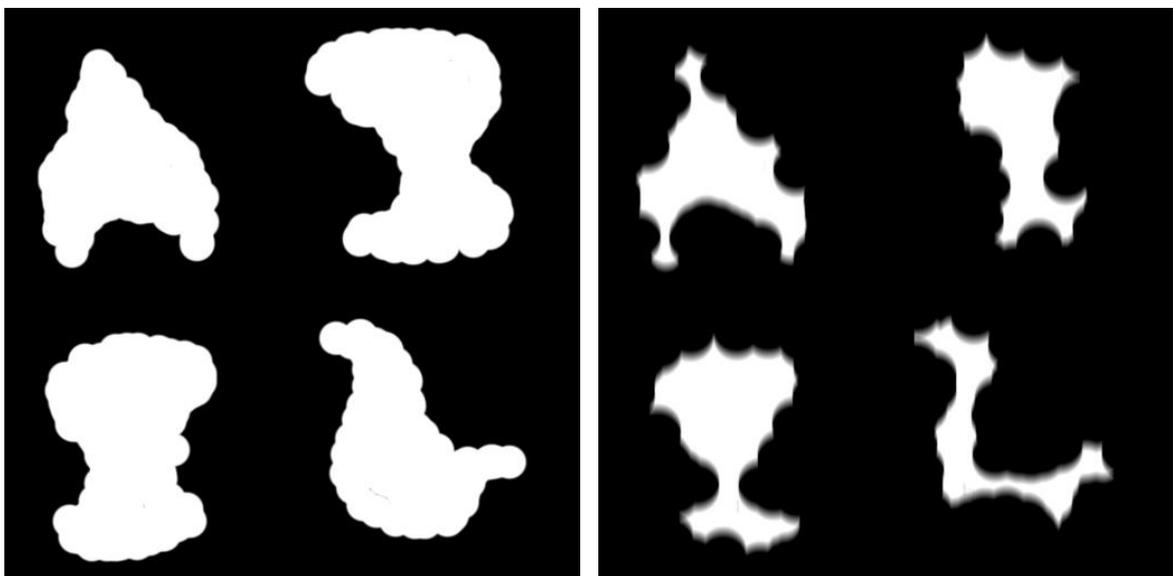
1. У полі **Hierarchy** створюється елемент **Particle System** шляхом ПКМ/Effect/Particle System.
2. У полі **Transform** задається розмір та положення площі, яку займатиме елемент.
3. Далі необхідно провести попереднє налаштування самого елемента.

Для цього проставити наступні параметри:

- **Start Lifetime** – 3;
- **Start Speed** – 4;
- **Start Color** – встановити сіруватий колір;
- **Start Size** – 2;
- **Shape** – встановити режим Cone та Angle = 15;

- **Emission** – обрати даний пункт та поставити значення Rate Over Time  $\pm 160$ .

4. Наступним кроком необхідно створити текстуру для основи вогню. Для цього спочатку треба створити новий документ у Photoshop розміром 512x512 пікселів у форматі RGB 72dpi. Далі залити фон чорним, розділити навпіл по горизонталі та вертикалі за допомогою направляючих та створити новий шар. На цьому шарі створити 4 елементи (кожний у своїй четверті) за допомогою пензлика із розміром Opacity 100% та розміром 30 пікселів. Має вийти щось схоже на приклад, наведений на рисунку 3.3.5 а. Потім необхідно взяти гумку з розміром 40 пікселів та дещо підтерти ці елементи, щоб вони були схожі на частинки полум'я. Має вийти щось схоже на приклад на рисунку 3.3.5 б. Далі треба застосувати ефект Motion Blur (Розмиття у русі) із значенням кута 90. Зберегти отриманий файл у форматі .jpeg та втягнути цей файл у поле **Project** у новостворену папку у папці **Asset**, вибрати його та поставити режим у полі **Texture Type** «**Sprite 2D and UI**», натиснути **Apply**;



а)

б)

Рисунок 3.3.5 – Приклад текстури

5. Створити у цій же папці **Material**, поставити у полі **Inspector** у полі **Shader** обрати **Particles/Standard Unlit** та у полі **Render Mode** режим

**Additive**, перетягнути створене у попередньому пункті зображення на поле **Albedo** та перетягнути даний матеріал у поле **Renderer/Material** поля **Inspector** елемента **Particle System**, створеного у попередніх пунктах;

6. Далі треба налаштувати зовнішній вигляд частинок вогню, аби це не був просто квадрат з 4 елементами. Для цього треба налаштувати:

- **Texture Sheet Animation** – підключити і встановити значення Tiles X=2, Y=2;

- **Frame over Time** – від 0 до 3,9996 (таким чином обирається наблизений зовнішній вигляд сніжинки відповідно до створеного у 5 пункті елемента);

7. Далі слід налаштувати параметр **Size over lifetime** – обрати цей пункт, клацнути на полі **Size** та сформувати криву, схожу на представлену на рисунку 3.3.6, а також поставити початкове значення 2 у полі редагування кривої у лівому верхньому куті. Це дозволить зробити зміну розміру більш плавною та реалістичною;



Рисунок 3.3.6 – Крива зміни розміру

8. **Color over lifetime** – обрати цей пункт, клацнути на білому полі **Color**, встановити мітки 10% та 90% на верхній частині діапазону та встановити значення Alpha 0 для 0% та 100%. Тоді слід налаштувати колір. Має бути 3 бігунки у нижній частині діапазону – жовтий, рудий та червоний. Зайві можна видалити.

9. Обрати параметр **Start Rotation** та встановити **Random Between Two Constants** і значення від -15 до 50 (це дозволить уникнути повторів елементів один за одним).

10. Далі треба створити джерело вогню. Натискаємо ПКМ на **Particle System** у полі **Hierarchy** та знову обираємо Effect/Particle System.

11. Необхідно провести попереднє налаштування самого елемента. Для цього проставити наступні параметри:

- **Start Lifetime** – 1,2;
- **Start Speed** – 2,5;
- **Start Color** – встановити темно-сірий колір;
- **Start Size** – 2;
- **Shape** – встановити режим Cone та Angle = 25, Radius = 0.2;
- **Emission** – обрати даний пункт та поставити значення Rate Over Time  $\pm 20$ ;
- Скопіювати колір з основного **Particle System** і вставити його до того ж самого поля цього дочірнього.

12. Наступним кроком необхідно створити текстуру для світіння джерела. Для цього спочатку треба створити новий документ у Photoshop розміром 100x100 пікселів у форматі RGB 72dpi. Далі залити фон чорним, розділити навпіл по горизонталі та вертикалі за допомогою направляючих та створити новий шар. На цьому шарі створити за допомогою інструменту градієнт (обрати радіальний та у виборі кольору зробити його білим з прозорою підкладкою), провівши ним від центру прямо праворуч до межі зображення. Задати Opacity шару 50%. Потім створити новий шар, виділити область за допомогою круглого інструменту для виділення та за допомогою інструменту градієнт з тими ж параметри створити радіальний градієнт, провівши їм від центру прямо праворуч. Далі створити новий шар та створити в центрі коло за допомогою еліпса із рамкою та заливкою білого кольору. Зберегти отриманий файл у форматі .jpeg та втягнути цей файл у поле **Project** у новостворену папку у папці **Asset**, вибрати його та поставити режим у полі **Texture Type** «**Sprite 2D and UI**», натиснути **Apply**;

13. Створити у цій же папці **Material**, поставити у полі **Inspector** у полі **Shader** обрати **Particles/Standard Unlit** та у полі **Render Mode** режим **Additive**, перетягнути створене у попередньому пункті зображення на поле **Albedo** та перетягнути даний матеріал у поле **Renderer/Material** поля **Inspector** елементу **Particle System**, створеного у пункті 10.

14. Трошки посунути елемент нижче, аби здавалося, що це основа вогню.

15. Далі слід налаштувати параметр **Size over lifetime** – обрати цей пункт, клацнути на полі **Size** та сформувати криву, схожу на представлену на рисунку 8, а також поставити початкове значення 2 у полі редагування кривої у лівому верхньому куті.

16. Створити дим. Для цього скопіювати основний **Particle System**, що був створений першим та видалити дочірній елемент у ньому. Перетягнути його на основний, зробивши його дочірнім. Потім змінити деякі параметри, а саме:

- **Start Lifetime** – 3;
- **Start Speed** – 4;
- **Start Color** – встановити темно-сірий колір (майже чорний);
- **Start Size** – 2;
- **Shape** – встановити режим Cone та Angle = 30, Radius = 1;
- **Order in Layer** (Renderer) – встановити значення -1, а у основного **Particle System** – встановити значення 1;
- У основного **Particle System** встановити режим **By Distance** у полі **Sort Mode** (Renderer).

17. Остання частина – створення іскор. Для цього треба спочатку натиснути ПКМ на **Particle System** у полі **Hierarchy** та знову обрати **Effect/Particle System**. Потім проставити наступні параметри:

- **Start Lifetime** – 5;
- **Start Speed** – 2;
- **Start Color** – встановити колір рудий основного елементу зі значенням альфа-каналу 80;

- **Start Size** – від 0,1 до 0,2;
- **Shape** – встановити режим Cone та Angle = 33, Radius = 0.3;
- **Emission** – обрати даний пункт та поставити значення Rate Over Time  $\pm 30$ .
- Підключити параметр **Noise**.

18. Перетягнути створений у пунктах 12-13 матеріал у поле **Renderer/Material** поля **Inspector** елемента **Particle System**, створеного у пункті 17.

Загалом можна ще позмінювати попередні параметри. У результаті буде отриманий ефект вогню, який можна підв'язати до маркера шляхом перетягування **Particle System** на маркер у полі **Hierarchy**

#### 4. Ефект входження у потік **Wrap Mode**

1. У полі **Hierarchy** створюється елемент **Particle System** шляхом ПКМ/Effect/Particle System.

2. У полі **Transform** задається розмір та положення площі, яку займатиме елемент.

3. Далі необхідно провести попереднє налаштування самого елемента. Для цього проставити наступні параметри:

- **Start Lifetime** – 2;
- **Start Speed** – 0;
- **Start Color** – встановити Gradient та проставити різні кольори;
- **Start Size** – від 0,15 до 0,25;
- **Shape** – встановити режим Cone, Angle = 60, Radius = 4, Radius Thickness = 0 (аби частинки генерувалися по краях площі системи), Emit From = Volume, Length = 3, Rotation = -90 (аби частинки летіли на спостерігача у камеру);
- **Emission** – обрати даний пункт та поставити значення Rate Over Time  $\pm 100$ ;
- **Velocity over Lifetime** – для швидкості по осі Y поставити значення  $\pm 90$ .

4. Наступним кроком необхідно створити текстуру для основи елемента потоку. Для цього спочатку треба створити новий документ у Photoshop розміром 100x100 пікселів у форматі RGB 72dpi. Далі залити фон чорним, розділити навпіл по горизонталі за допомогою направляючих. Створити коло за допомогою еліпса та відкласти від нього 2 прями, що закінчуються в одній точці. Зафарбувати простір між колом та прямими, растеризувавши одну з них. Має вийти щось схоже на приклад на рисунку 3.3.7. Зберегти отриманий файл у форматі .jpeg та втягнути цей файл у поле **Project** у новостворену папку у папці **Asset**, вибрати його та поставити режим у полі **Texture Type** «**Sprite 2D and UI**», натиснути **Apply**;

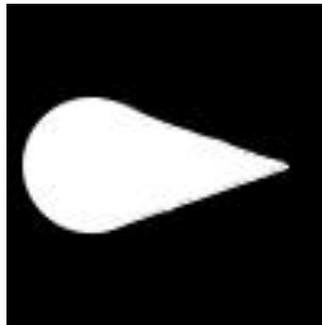


Рисунок 3.3.7 – Приклад текстури

5. Створити у цій же папці **Material**, поставити у полі **Inspector** у полі **Shader** обрати **Particles/Standard Unlit** та у полі **Render Mode** режим **Additive**, перетягнути створене у попередньому пункті зображення на поле **Albedo** та перетягнути даний матеріал у поле **Renderer/Material** поля **Inspector** елемента **Particle System**, створеного у попередніх пунктах.

6. Провести певне налаштування частинок потоку, а саме:

- **Renderer/Renderer Mode** – Stretched Billboard;
- **Renderer/Length Scale** – встановити значення 25.

Загалом можна ще позмінювати попередні параметри. У результаті буде отриманий ефект входження у потік Wrap Mode, який можна підв'язати до маркера шляхом перетягування **Particle System** на маркер у полі **Hierarchy**

## 5.Ефект магичної руни

1. Завантажити у середовище ПЗ Photoshop зображення руни, схожій на приклад на рисунку 3.3.8, та зробити його інверсією Image/Adjustments/Invert. Зберегти отриманий файл у форматі .jpeg та втягнути цей файл у поле **Project** у новостворену папку у папці **Asset**, вибрати його та поставити режим у полі **Texture Type** «**Sprite 2D and UI**», натиснути **Apply**.



Рисунок 3.3.8 – Приклад зображення для руни

2. У полі **Hierarchy** створюється елемент **Particle System** шляхом ПКМ/Effect/Particle System.

3. Створити у папці з пункту 1 **Material**, поставити у полі **Inspector** у полі **Shader** обрати **Particles/Standard Units** та у полі **Render Mode** режим **Additive**, перетягнути створене у попередньому пункті зображення на поле **Albedo** та перетягнути даний матеріал у поле **Renderer/Material** поля **Inspector** елемента **Particle System**, створеного у попередньому пункті.

4. У полі **Transform** задається розмір та положення площі, яку займатиме елемент.

5. Провести наступне налаштування:

- **Start Lifetime** – 3;
- **Start Speed** – 0;
- **Start Color** – встановити будь-який колір;
- **Start Size** – 3;
- **Shape** – прибрати;
- **Emission** – обрати даний пункт та поставити значення Rate Over Time 1
- **Max Particles** – 1;
- **Color over lifetime** – обрати цей пункт, клацнути на білому полі Color, встановити мітки 10% та 90% на верхній частині діапазону та встановити значення Alpha 0 для 0% та 100%;
- Підрегулювати **Size over Lifetime** для більшої плавності анімації;
- **Rotation by Speed** – обрати та поставити значення  $\pm 60$ ;
- **Renderer Mode** – Horizontal billboard;
- **Rendering/Max Particle size** – 3;

6. Далі треба створити пульсацію від руни. Для цього спочатку треба скопіювати попередній елемент в полі Hierarchy та зробити новостворений його дочірнім елементом, перетягнувши його на назву основного.

7. Далі треба створити анімацію пульсації для руни. Для цього спочатку треба створити новий документ у Photoshop розміром 2000x2000 пікселів у форматі RGB 72dpi. Далі залити фон чорним, розділити навпіл по горизонталі та вертикалі за допомогою направляючих. Створити коло за допомогою еліпса без заливки з білим контуром товщиною 15 пікселів та додати до нього ефект зовнішнього світіння, параметри якого представлені на рисунку 3.3.9. Зберегти отриманий файл у форматі .jpeg та втягнути цей файл у поле **Project** у новостворену папку у папці **Asset**, вибрати його та поставити режим у полі **Texture Type** «**Sprite 2D and UI**», натиснути **Apply**;

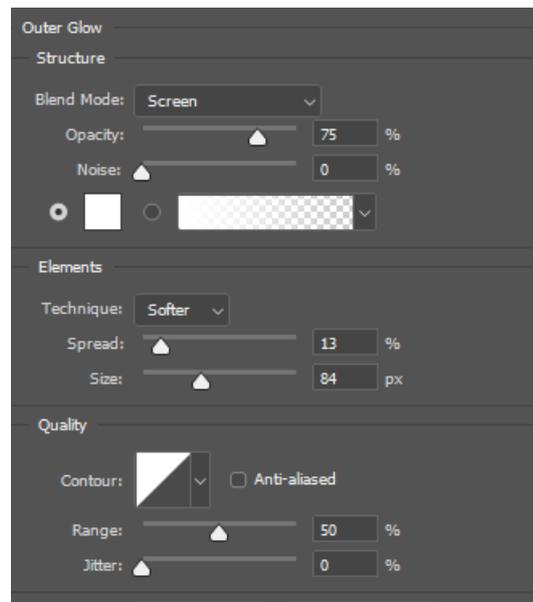


Рисунок 3.3.9 – Параметри зовнішнього світіння для текстури

8. Створити у папці з пункту 7 **Material**, поставити у полі **Inspector** у полі **Shader** обрати **Particles/Standard Unlit** та у полі **Render Mode** режим **Additive**, перетягнути створене у попередньому пункті зображення на поле **Albedo** та перетягнути даний матеріал у поле **Renderer/Material** поля **Inspector** елементу **Particle System**, створеного у пункті 6.

9. Налаштувати певні параметри:

- **Start Lifetime** – 3;
- **Start Speed** – 0;
- **Start Color** – прибрати;
- **Start Size** – 5;
- **Shape** – прибрати;
- **Emission** – обрати даний пункт та поставити значення Rate Over Time 10
- **Max Particles** – 10;
- **Color over lifetime** – обрати цей пункт, клацнути на білому полі Color, встановити мітки 10% та 90% на верхній частині діапазону та встановити значення Alpha 0 для 0% та 100%, а також можна створити певний градієнт для зафарбовування кілець;

- Підрегулювати **Size over Lifetime** для більшої плавності анімації;
- **Rotation by Speed** – прибрати;
- **Renderer Mode** – Horizontal billboard

10. Далі можна створити анімацію кілець вгору від руни. Для цього слід скопіювати попередній створений елемент, та провести певні налаштування:

- **Start Lifetime** – 2;
- **Start Speed** – 1;
- **Start Color** – прибрати;
- **Start Size** – 3;
- **Simulation Speed** – 0.8;
- **Shape** – прибрати;
- **Emission** – обрати даний пункт та поставити значення Rate Over Time 4;
- **Max Particles** – 25;
- **Color over lifetime** – обрати цей пункт, клацнути на білому полі Color, встановити мітки 10% та 90% на верхній частині діапазону та встановити значення Alpha 0 для 0% та 100%, а також можна створити певний градієнт для зафарбовування кілець;

11. Також можна створити дим над руною. Для цього створити новий елемент **Particle System** та зробити його дочірнім основного.

12. Для цього спочатку створити новий документ у Photoshop розміром 128x128 пікселів у форматі RGB 72dpi. Далі залити фон чорним, взяти пензлик з Opacity 30% та розміром 10 пікселів та намалювати щось схоже на приклад на рисунку 3.3.10. Зберегти отриманий файл у форматі .jpeg та втягнути цей файл у поле **Project** у новостворену папку у папці **Asset**, вибрати його та поставити режим у полі **Texture Type** «**Sprite 2D and UI**», натиснути **Apply**;

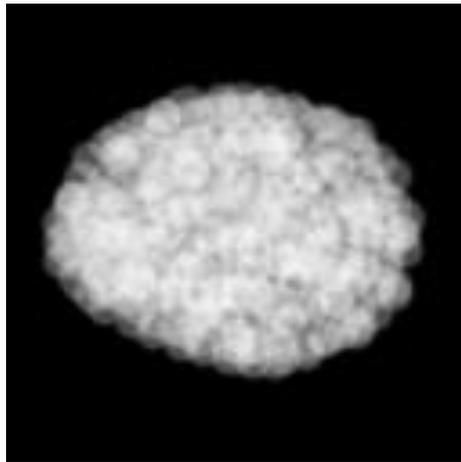


Рисунок 3.3.10 – Приклад текстури

13. Провести певні налаштування:

- **Start Lifetime** – 5;
- **Start Speed** – 0,25;
- **Start Color** – зробити альфа такою, аби дим був прозорий;
- **Start Size** – 0,5;
- **Start Rotation** – від 0 до 180;
- **Simulation Speed** – 0.8;
- **Shape** – обрати режим Cone та Angle=70;
- **Emission** – обрати даний пункт та поставити значення Rate Over Time  $\pm 100$ ;
- **Max Particles** – 1000;
- **Color over lifetime** – обрати цей пункт, клацнути на білому полі Color, встановити мітки 10% та 90% на верхній частині діапазону та встановити значення Alpha 0 для 0% та 100%, а також можна створити певний градієнт для зафарбовування диму.

14. Також можна створити частинки, що вилітають з руни. Для цього знову ж таки створюється новий дочірній елемент **Particle System**. Потім підвантажуються текстура (див. п.12 попереднього ефекту) та налаштовуються певні параметри, а саме:

- **Start Lifetime** – 5;

- **Start Speed** – 1;
- **Start Color** – зробити альфа такою, аби частинки були прозорі;
- **Start Size** – 0,05;
- **Shape** – обрати режим Cone та Angle=0;
- **Emission** – обрати даний пункт та поставити значення Rate Over Time  $\pm 10$ ;
- **Max Particles** – 1000;
- **Color over lifetime** – обрати цей пункт, клацнути на білому полі Color, встановити мітки 10% та 90% на верхній частині діапазону та встановити значення Alpha 0 для 0% та 100%, а також можна створити певний градієнт для зафарбовування частинок;
- **Rendering/Max Particle Size** – 0.5;
- **Renderer Mode** – Billboard

Загалом можна ще позмінювати попередні параметри. У результаті буде отриманий ефект магічної руни, який можна підв'язати до маркеру шляхом перетягування **Particle System** на маркер у полі **Hierarchy**

### Хід роботи

1. Створити ефект, який сюжетно підходить під створений додаток комп'ютерного практикуму №2 (можна з представлених у прикладі, а можна й свій).
2. Підв'язати його під маркер та створити новий додаток.
3. Елемент доповненої реальності повинен мати якусь логіку та ідею.
4. Створити звіт з виконаної роботи з супровідними скріншотами.

### Контрольні запитання

1. Особливості ефекту Particle System.
2. Параметри та модулі ефекту Particle System.

## КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 3.4

### СТВОРЕННЯ МЕНЮ ТА ПЕРЕХОДІВ МІЖ СЦЕНАМИ ДОДАТКУ З ЕЛЕМЕНТАМИ ДОПОВНЕНОЇ РЕАЛЬНОСТІ ЗА ДОПОМОГОЮ UNITY

**Мета:** Створити додаток із меню та декількома сценами та створити можливість переходів між ними

#### Теоретичні відомості

##### 1. Створення кнопок

Середовище Unity дозволяє створювати кнопки, які дозволятимуть здійснювати різноманітні запрограмовані дії. Наприклад, перехід на наступну сцену, повернення до попередньої, вихід з гри тощо. Для створення кнопки необхідно спочатку створити новий об'єкт **UI/Button** у полі **Hierarchy**. Далі за допомогою параметрів (**Rect Transform**, **Source Image** тощо) поля **Inspector** можна налаштувати положення, розмір, вирівнювання кнопок, їх відступи від країв, підписи, зовнішній вигляд тощо. Для видозміни зовнішнього вигляду кнопок слід підвантажити необхідні зображення та задати **Texture Type Sprite (2D and UI)**. Далі треба створити порожній об'єкт, обрати його, натиснути на кнопку **Add Component** у полі **Inspector** та обрати **New Script**, зайти у нього та написати код, представлений у таблиці 2.

Далі після написання необхідного коду для створення кнопок переходу між сценами або ж виходу з додатку необхідно виконати наступне:

1. Встановити в параметрі **Button** поля **Inspector** тип **Transition/Animation** для кожної кнопки.

2. У кожній кнопці у параметрі **On click** створити новий стан та перетягнути у порожнє поле створений порожній об'єкт з кодом та обрати у полі «**No function**» назву створеного методу (**LoadScenes**), а у ньому метод **ChangeScene ()**.

3. Створити декілька сцен та їх наповнення (**Project/Scenes/ПКМ/Create/Scene**).

4. Завантажити усі сцени до **Hierarchy**.

5. Перейти **File/Build Settings** та натиснути кнопку **Add open scenes**.

Розставити сцени так як потрібно та запам'ятати їх порядковий номер, який потім внести до поля **Inspector/OnClick** відповідної кнопки.

Для кнопки виходу з додатку порядок створення аналогічний, лише у 2 пункті слід обирати метод **ExitPressed**.

## 2. Створення контролеру руху моделі

Також можливим є задання певного контролеру для управління 3D-моделлю. Для цього треба завантажити Standard Assets з Інтернету та імпортувати даний пакет в Unity. Потім треба створити меню контролера шляхом витягування його за наступним шляхом **Project/Standard Assets/CrossPlatformInput/Prefabs/MobileSingleStickControl**. Далі треба видалити зі створеного елемента кнопку, що відповідає за анімацію стрибка персонажа (якщо такої немає). Також треба підвантажити до поля **Project** анімацію ходіння персонажа та стрибка (якщо потрібно). Далі треба натиснути на кнопку **Add Component** у полі **Inspector** та обрати **New Script**, зайти у нього та написати код, представлений у таблиці 2. Потім для кожної анімації обрати у полі **Inspector/Rig/Animation Type/Legacy/Apply**. Далі треба обрати дану анімацію, натиснути **Edit** в полі **Inspector**, перейменувати так, як дана анімація називається у коді, та зациклити її шляхом **Wrap Mode/Loop**. Далі треба обрати саму модель, у полі **Inspector/Animation** поставити 2 та перетягнути 2 анімації у порожні місця цього параметру.

Після налаштування усіх параметрів необхідно задати код для того, щоб кнопка виконувала поставлені задачі. Приклад коду наведений у таблиці 3.4.1.

Таблиця 3.4.1 – Приклад кодів для управління кнопками

Рядок коду	Характеристика, за яку відповідає
Управління контролером	
<code>using UnityEngine;</code>	Підключення основної бібліотеки Unity
<code>using UnityStandardAssets.CrossPlatformInput;</code>	Підключення бібліотеки для управління контролером
<code>public class SansScript : MonoBehaviour</code> {	
<code>private Animation anim;</code>	Задання змінної anim, що відповідатиме за анімацію моделі
<code>private Rigidbody rb;</code>	Задання змінної rb, що відповідатиме за підключення параметрів Rigidbody
<code>[SerializeField]</code>	Характеристика, що дозволяє приватну змінну показувати в полі Inspector
<code>private float speed = 4f;</code>	Задання змінної speed, що відповідатиме за швидкість
<code>void Start()</code> {	Опис методу Start
<code>rb = GetComponent&lt;Rigidbody&gt;();</code>	Присвоєння компоненту Rigidbody змінній rb
<code>anim = GetComponent&lt;Animation&gt;();</code>	Присвоєння компоненту Animation змінній anim
}	
<code>void Update()</code> {	Опис методу Update
<code>float x =</code> <code>CrossPlatformInputManager.GetAxis("Horizontal");</code>	Задання змінної для руху контролера по осі X
<code>float y =</code> <code>CrossPlatformInputManager.GetAxis("Vertical");</code>	Задання змінної для руху контролера по осі Y
<code>Vector3 movement = new Vector3(x, 0, y);</code>	Присвоєння руху моделі
<code>rb.velocity = movement * speed;</code>	Швидкість руху моделі
<code>if (x != 0 &amp;&amp; y != 0)</code> <code>transform.eulerAngles = new</code> <code>Vector3(transform.eulerAngles.x, Mathf.Atan2(x,</code> <code>y) * Mathf.Rad2Deg, transform.eulerAngles.z);</code>	Підлаштування кута повороту моделі при русі під напрямком руху контролера
<code>if (x != 0    y != 0)</code> <code>anim.Play("Walking");</code>	Задання анімації моделі – якщо модель рухається, то програти анімацію під назвою <b>Walking</b> у даному випадку, якщо стоїть – то <b>Dance</b>
<code>else</code>	
<code>anim.Play("Dance");</code>	
}	
}	
Перехід між сценами	
<code>using System.Collections;</code>	Підключення бібліотек
<code>using System.Collections.Generic;</code>	
<code>using UnityEngine;</code>	
<code>using UnityEngine.SceneManagement;</code>	
<code>public class LoadScenes : MonoBehaviour</code> {	
<code>// Start is called before the first frame update</code>	

Кінець таблиці 3.4.1

Рядок коду	Характеристика, за яку відповідає
<code>public void ChangeScenes(int _number)</code>	Задання методу ChangeScenes
<code>{</code>	
<code>SceneManager.LoadScene(_number);</code>	Перехід на сцену з певним номером
<code>}</code>	
<code>}</code>	
Вихід з додатку	
<code>using System.Collections;</code>	Підключення бібліотек
<code>using System.Collections.Generic;</code>	
<code>using UnityEngine;</code>	
<code>using UnityEngine.SceneManagement;</code>	
<code>public class exitgame : MonoBehaviour</code>	
<code>{</code>	
<code>public void ExitPressed()</code>	Задання методу ExitPressed
<code>{</code>	
<code>Application.Quit();</code>	Вихід з системи
<code>Debug.Log("Exit pressed!");</code>	Необов'язковий рядок, але він дасть змогу перевірити правильність коду в меню Unity
<code>}</code>	
<code>}</code>	

### Хід роботи

1. Створити меню для додатку з елементами доповненої реальності, створеного раніше з кнопками переходу на різні сцени (включаючи сцену з доповненою реальністю).
2. Створити кнопки повернення до меню.
3. Створити контролер для управління рухом моделі.
4. Підв'язати контролер під маркер.
5. Створити звіт з виконаної роботи з супровідними скріншотами.

### Контрольні питання

1. Алгоритм створення переходу між сценами.
2. Алгоритм створення кнопок.
3. Алгоритм створення контролеру для управління персонажами.

## КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 3.5

### РОЗРОБЛЕННЯ МАРКЕРІВ ДОПОВНЕНОЇ РЕАЛЬНОСТІ ТА ВИЗНАЧЕННЯ ВПЛИВУ ХАРАКТЕРИСТИК ЗАДРУКОВУВАНОВОГО МАТЕРІАЛУ НА ЇХ ПОЛІГРАФІЧНЕ ВІДТВОРЕННЯ ТА ВІДТВОРЕННЯ НА ЕКРАНІ ЦИФРОВОГО КОНТЕНТУ

**Мета:** Створити маркер доповненої реальності та визначити вплив характеристик матеріалу на їх поліграфічне виконання.

#### Теоретичні відомості

**Маркер доповненої реальності** – це основа для використання маркерної технології. Це чорно-біле або кольорове просте або деталізоване зображення, при наведенні на яке камери скануючого пристрою відбувається відтворення віртуальних об'єктів. Маркер може бути надрукований або ж розміщений в електронному вигляді на сторінці сайту. Маркери бувають різні.

Одним із найпоширеніших є **простий маркер**, оскільки його легко створювати та вводити у контент реальної продукції. Є ряд характеристик, яким має відповідати такого роду маркер: він має бути чорно-білим, інваріантним до поворотів, квадратним та мати товсту чорну рамку, яка займатиме 10% його площі. Основні алгоритми розпізнавання, застосовувані у даному випадку – це розпізнавання кутів, країв, точок та контурів. Цей вид маркеру має найбільші показники стабільності та легкості при використанні.

Наступний найбільш застосовуваний вид маркеру є так званий «**безмаркерний**» тип. Не дивлячись на назву маркер все-таки є – це деталізоване зображення. Воно має мати велику кількість деталей і бути дуже якісним, кольоровим та інваріантним до поворотів. Розпізнавання базується на зміні градієнту, розпізнавання кутів зображення або ж локаційних координатах. До основних причин широкого застосування можна віднести

легкість введення таких зображень у реальний світ через відсутність необхідності створення чорної рамки навколо зображення.

**Комбінація маркерів** застосовується для технології, що базується на розпізнаванні 3Д-об'єктів, наприклад для представлення продукції у реальних умовах. Тобто спочатку створюється 3Д-модель об'єкту, потім розробляється код, і при наведенні камери на реальний об'єкт відтворюється контент.

**Frame Marker** застосовується для шифрування логотипів чи слоганів компанії для їх індивідуальної ідентифікації. Такі маркери мають бути маленькими, інваріантними до поворотів, квадратними, уніфікованими та мати контраст між рамкою та наповненням.

До основних параметрів класифікації маркерів можна віднести наступні: основа для розпізнавання, метод розпізнавання, колір маркерів, складність рисунку маркеру, сфера використання тощо.

### Хід роботи

1. Створити тест-об'єкт, що міститиме прості маркери доповненої реальності (до 4 штук) зі зміною розміру для оцінювання чіткості їх відтворення, а також штрихове поле у горизонтальному напрямку зі зміною товщини ліній для оцінювання якості відтворення тонких ліній, їх спотворення на відбитках та якості країв.

2. Роздрукувати тест-об'єкт на декількох матеріалах з різними характеристиками.

3. Перевірити як характеристики матеріалу (колір, пористість, додаткове покриття, білість, глянець, непрозорість) впливатимуть на поліграфічне відтворення маркерів – товщину ліній, чорний колір та відповідність та якість відтворення тонких елементів та рисунку маркеру – за допомогою електронного мікроскопу та спектрофотометру.

4. Порівняти отримані значення з оригінальними, оцінити отримані відбитки візуально на наявність недоліків та зробити висновки стосовно впливу характеристик маркерів на їх поліграфічне виконання.

5. Підв'язати маркери під простий елемент доповненої реальності та перевірити як характеристики матеріалів впливають на відтворення маркерів на пристрої.

6. Створити протокол з результатами проробленої роботи.

### **Контрольні питання**

1. Маркер доповненої реальності.
2. Види маркерів та їх характеристики.
3. Класифікаційні характеристики маркерів.

### **Перелік літератури за темою**

1. Распознавание маркера дополненной реальности [Електронний ресурс] – Режим доступу: <https://habr.com/ru/post/135659/>, вільний. – Назва з екрану – Мова рос.

2. Создать свою дополненную реальность [Електронний ресурс] – Режим доступу: <https://theroar.io/sozdat-svoyu-dopolnennuyu-realnost/>, вільний. – Назва з екрану – Мова рос.

3. Живые 3D-метки [Електронний ресурс] – Режим доступу: <https://www.eligovision.ru/ru/know-how/markers/>, вільний. – Назва з екрану – Мова рос.

**РЕКОМЕНДОВАНИЙ ПЕРЕЛІК ЛІТЕРАТУРНИХ ДЖЕРЕЛ**

1. Яковлев Б. С. Классификация и перспективные направления использования технологии дополненной реальности [Текст] / Б. С. Яковлев, С. И. Пустов // Известия Тульского государственного университета. Технические науки. – 2013.
2. Технологии и алгоритмы для создания дополненной реальности [Электронный ресурс] – Режим доступа: <https://www.mais-journal.ru/jour/article/view/211/221>, вільний. – Назва з екрану – Мова рос.
3. Технология виртуальной и дополненной реальности [Электронный ресурс] – Режим доступа: <https://cyberleninka.ru/article/n/tehnologii-virtualnoy-i-dopolnennoy-realnosti-vozmozhnosti-i-prepyatstviya-primeneniya>, вільний. – Назва з екрану – Мова рос.
4. Технологии и алгоритмы для создания дополненной реальности [Электронный ресурс] – Режим доступа: <https://www.mais-journal.ru/jour/article/view/211/221>, вільний. – Назва з екрану – Мова рос.
5. Технологии дополненной реальности [Электронный ресурс] – Режим доступа: <http://dgng.pstu.ru/conf2017/papers/110/>, вільний. – Назва з екрану – Мова рос.
6. Дополненная реальность [Электронный ресурс] – Режим доступа: <http://old.tusur.ru/filearchive/reports-magazine/2010-1-2/161-165.pdf/>, вільний. – Назва з екрану – Мова рос.
7. Распознавание маркера дополненной реальности [Электронный ресурс] – Режим доступа: <https://habr.com/ru/post/135659/>, вільний. – Назва з екрану – Мова рос.
8. Создать свою дополненную реальность [Электронный ресурс] – Режим доступа: <https://theroar.io/sozdat-svoyu-dopolnennuyu-realnost/>, вільний. – Назва з екрану – Мова рос.
9. Живые 3D-метки [Электронный ресурс] – Режим доступа: <https://www.eligovision.ru/ru/know-how/markers/>, вільний. – Назва з екрану – Мова рос.

Міністерство освіти і науки України  
Національний технічний університет України  
«Київський політехнічний інститут ім. І. Сікорського»  
Видавничо-поліграфічний інститут

РОБОТА №\_\_ на тему:

«\_\_\_\_\_»

Виконав:

Студент групи \_\_\_\_\_

Перевірив: \_\_\_\_\_

доц., к.т.н. \_\_\_\_\_