

# Лек

## **XR-технології на практиці: створення іммерсивного додатку з використанням Unity**

Викладач Ph.D, в.о. зав.каф. КН Граф М.С.

# План

1. Створення **AR-додатку** для смартфонів з підтримкою Google AR на платформі Android.
2. Створення **VR-додатку**
3. Створення **MR-додатку**

Для реалізації XR-технологій нам допоможуть такі готові набори інструментів:

**AR Foundation**

**XR Interaction Toolkit**

У результаті в нас вийде три додатки:

**AR-програма**

**VR-додаток**

**MR-додаток**

Примітка: всі програми збиратимемо до одного Unity-проєкту версії 2022.3.29f1, що зарекомендувала себе як стабільна при використанні пакетів AR Foundation і XR Interaction Toolkit.

# 1. Створення AR-додатку для смартфонів з підтримкою Google AR на платформі Android.

Після запуску редактора відкрийте диспетчер пакетів через **Add package by name** та встановіть версію AR Foundation 5.1.5 (ім'я пакета: `com.unity.xr.arfoundation`).

Після встановлення пакета налаштуйте сцену:

1. Клацніть правою кнопкою миші (ПКМ) в області перегляду сцени. У меню, що випадає, виберіть XR → AR Session. На сцену додасться об'єкт із компонентами AR Session та AR Input Manager, які керують AR-сесією.
2. Так само створіть XR Origin для роботи з AR-камерою. Виберіть XR → XR Origin (Mobile AR). На об'єкті Main Camera автоматично з'являться компоненти AR Camera Manager, AR Camera Background та Tracked Pose Driver для керування камерою.

# 1. Створення AR-додатку для смартфонів з підтримкою Google AR на платформі Android.

Виявлення площин за допомогою компонентів AR Plane Manager та AR Point Cloud Manager:

- У компоненті AR Plane Manager у полі Detection Mode виберіть значення Horizontal, щоб знаходити горизонтальні площини. У полі Plane Prefab вкажіть префаб площини, який відобразиться, щоб користувач розумів, що цю область можна використовувати для розміщення об'єктів.
- Створіть префаб плоскості. Для цього натисніть ПКМ в області перегляду сцени та виберіть XR → AR Default Plane. Налаштуйте об'єкт, вибравши відповідний колір матеріалу в компоненті Mesh Renderer, а також колір та товщину ліній у Line Renderer. На AR Plane налаштуйте поле Vertex Changed Threshold. Збільшення значення підвищує продуктивність, але знижує точність і навпаки. Налаштування залежать від ваших ресурсів та потреб. Готовий об'єкт перемістіть до папки Prefabs та додайте посилання на нього у полі Plane Prefab компонента AR Plane Manager.
- Створіть префаб для AR Point Cloud. Аналогічно виберіть XR → AR Default Point Cloud. Цей об'єкт містить компоненти AR Point Cloud та AR Point Cloud Particle Visualizer, які забезпечують візуальне відображення частинок під час сканування поверхні. Налаштуйте об'єкт, збережіть як префаб і додайте до поля AR Point Cloud Manager.

# 1. Створення AR-додатку для смартфонів з підтримкою Google AR на платформі Android.

## Реалізація логіки розміщення об'єкта на поверхні

Створіть папку Scripts і в ній створимо MonoBehaviour-скрипт з назвою PlacerObjectsOnPlane. ARRaycastManager – позначити як обов'язковий компонент.

Отримуємо посилання на нього в методі Awake():

```
[RequireComponent(typeof(ARRaycastManager))]  
public class PlacerObjectsOnPlane : MonoBehaviour {  
    private ARRaycastManager _raycastManager;  
    private void Awake() => _raycastManager = GetComponent<ARRaycastManager>();  
}
```

# 1. Створення AR-додатку для смартфонів з підтримкою Google AR на платформі Android.

метод розміщення об'єкта:

```
public UnityEvent OnPlacedObject;
[SerializeField] private GameObject _placedPrefab;
private GameObject _spawnedObject;

private void Placement(Vector2 touchPosition) {
    List<ARRaycastHit> hits = new List<ARRaycastHit>();
    if (_raycastManager.Raycast(touchPosition, _hits, TrackableType.PlaneWithinPolygon)){
        Pose hitPose = _hits[0].pose;
        if(_spawnedObject == null){
            _spawnedObject = Instantiate(_placedPrefab, hitPose.position, hitPose.rotation);
            OnPlacedObject?.Invoke();
        }else _spawnedObject.transform.SetPositionAndRotation(hitPose.position, hitPose.rotation)
    }
}
```

# 1. Створення AR-додатку для смартфонів з підтримкою Google AR на платформі Android.

Тепер додамо логіку, яка оброблятиме натискання на екран.

Для цього реалізуємо метод Update:

```
private void Update() {  
    if (Input.touchCount > 0){  
        Touch touch = Input.GetTouch(0);  
        if (touch.phase == TouchPhase.Began && IsClickedOnUi(touch) == false) Placement(touch.position);  
    }  
}
```



# 1. Створення AR-додатку для смартфонів з підтримкою Google AR на платформі Android.

Для перевірки натискання на UI додамо наступний метод:

```
private bool IsClickedOnUi(Touch touch) {  
    PointerEventData eventDataCurrentPosition = new PointerEventData(EventSystem.current);  
    eventDataCurrentPosition.position = touch.position;  
    List<RaycastResult> results = new List<RaycastResult>();  
    EventSystem.current.RaycastAll(eventDataCurrentPosition, results);  
    foreach (var item in results) {  
        if (item.gameObject.CompareTag(TAG_UI)) return true;  
    }  
    return false;  
}
```

# 1. Створення AR-додатку для смартфонів з підтримкою Google AR на платформі Android.

Налаштування та збирання проєкту:

1. Розмістіть скрипт `PlacerObjectsOnPlane` на об'єкті `XROrigin`. При цьому автоматично буде додано компонент `ARRaycastManager`.
2. У полі `RaycastPrefab` компонента `ARRaycastManager` залиште значення порожнім.
3. У полі `_placedPrefab` компонента `PlacerObjectsOnPlane` вкажіть префаб вашого об'єкта.

Збирання проєкту під Android. Для цього:

1. Перейдіть в `File->Build Settings` і виберіть `Android` як платформу.
2. Додайте сцену до списку `Scenes in Build`.
3. У `Project Settings -> XR Plug-in Management` активуйте `ARCore` на вкладці `Android`.
4. У `Player -> Other Settings` вкажіть наступні параметри:
  - Заберіть галочку `Auto Graphics API` і залиште лише `OpenGL`.
  - Встановіть `Texture Compression Format` у значення `ASTC`.
  - Увімкніть тільки архітектуру `ARM64` у розділі `Target Architectures`.
  - Переконайтеся, що `Minimum API Level` не нижче `24`.
5. Підключіть тестовий пристрій із активованим режимом розробника та налагодженням.
6. Виберіть пристрій у полі `Run Device` і натисніть `Build And Run`, щоб скласти та запустити програму.

# 1. Створення AR-додатку для смартфонів з підтримкою Google AR на платформі Android.

Додавання примітивного UI

Canvas-> EventSystem->ReplaceWithInputModule

У компоненті Canvas Scaler:

1. У полі UI Scale Mode виберіть Scale With Screen Size.
2. У полі Reference Resolution встановіть X: 1080 та Y: 1920 (для портретної орієнтації).
3. У полі Screen Match Mode виберіть Match Width Or Height.
4. У полі Match встановіть значення 0.5, щоб забезпечити однаковий вплив ширини та висоти масштабування.

# 1. Створення AR-додатку для смартфонів з підтримкою Google AR на платформі Android.

Перемикач поверхонь скрипт PlacerSwitcher

```
[RequireComponent(typeof(ARPlaneManager), typeof(ARPointCloudManager), typeof(PlacerObjectsOnPlane))]
public class PlacerSwitcher : MonoBehaviour {
    [SerializeField] private Button _buttonForFixingObject;
    private ARPlaneManager _arPlaneManager;
    private PlacerObjectsOnPlane _placeObjectsOnPlane;
    private ARPointCloudManager _arPointCloudManager;

    private void Awake() {
        _arPlaneManager = GetComponent<ARPlaneManager>();
        _placeObjectsOnPlane = GetComponent<PlacerObjectsOnPlane>();
        _arPointCloudManager = GetComponent<ARPointCloudManager>();
    }

    private void OnEnable() {
        _placeObjectsOnPlane.OnPlacedObject.AddListener(ShowConfirmPlaceObjectButton);
        _buttonForFixingObject.onClick.AddListener(HideARPlane);
    }
}
```

# 1. Створення AR-додатку для смартфонів з підтримкою Google AR на платформі Android.

Перемикач поверхонь скрипт PlacerSwitcher (продовження)

```
private void OnDisable() {
    _placeObjectsOnPlane.OnPlacedObject.RemoveListener(ShowConfirmPlaceObjectButton);
    _buttonForFixingObject.onClick.RemoveListener(HideARPlane);
}

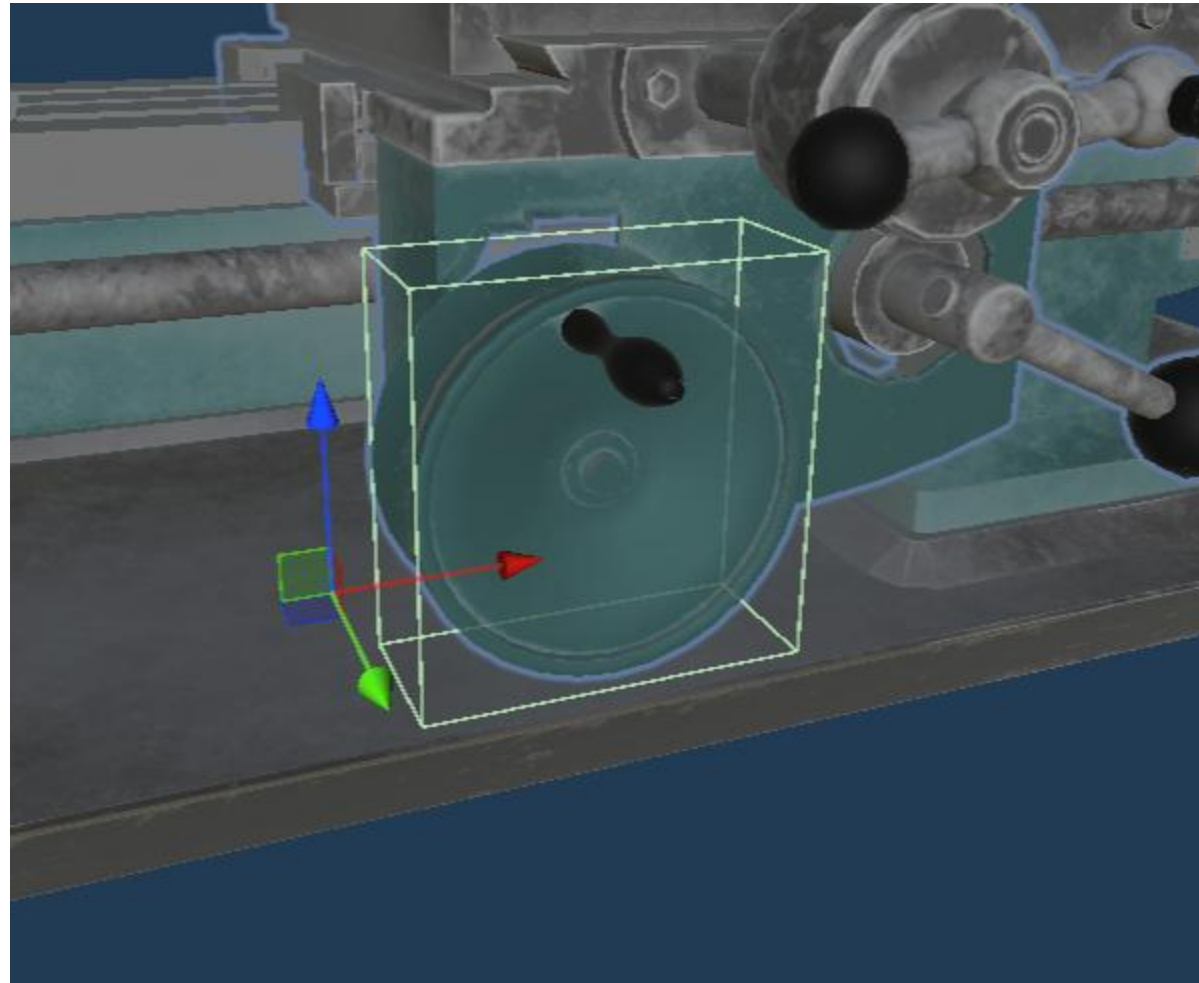
private void Start() => _buttonForFixingObject.gameObject.SetActive(false);
private void ShowConfirmPlaceObjectButton() => _buttonForFixingObject.gameObject.SetActive(true);

private void HideARPlane() {
    _placeObjectsOnPlane.enabled = false;
    _arPlaneManager.SetTrackablesActive(false);
    _arPointCloudManager.SetTrackablesActive(false);
    _arPlaneManager.enabled = false;
    _arPointCloudManager.enabled = false;
    _buttonForFixingObject.gameObject.SetActive(false);
}
```

```
}
```

# 1. Створення AR-додатку для смартфонів з підтримкою Google AR на платформі Android.

Переміщення платформи



# 1. Створення AR-додатку для смартфонів з підтримкою Google AR на платформі Android.

Переміщення платформи (скрипт MoverAlongAxis)

```
public class MoverAlongAxis : MonoBehaviour, IPointerDownHandler, IDragHandler
```

Щоб користувач міг вибрати, уздовж якої осі рухатиметься об'єкт, створимо enum Axis у файлі Params:

```
public enum Axis{ x = 0, y = 1 , z = 2}
```

Позначимо поле `_axis` типу enum Axis, описаного нами раніше, – це дозволить вибрати напрямок, вздовж якого об'єкт переміщатиметься і використовуватиме подальших розрахунках:

```
[SerializeField] private Axis _axis;
```

1. Створення **AR-додатку** для смартфонів з підтримкою Google AR на платформі Android.



Дякую за увагу!