

1. Przygotowanie projektu:

- a. Zainstaluj AR Foundation i ARKit/ARCore:
 - i. **AR Foundation** (wersja stabilna, np. 5.0+).
 - ii. **ARCore XR Plugin** (dla Androida).
 - iii. **ARKit XR Plugin** (dla iOS)
- b. Konfiguracja XR Plug-in Management:
 - i. Przejdź do **Edit > Project Settings > XR Plug-in Management**.
 - ii. Włącz **Plug-in Providers**:
 1. Dla Androida: zaznacz **ARCore**.
 2. Dla iOS: zaznacz **ARKit**.
- c. Ustawienia platformy:
- d. Android:
 - i. **File > Build Settings**, wybierz "Android".
 - ii. Kliknij "Switch Platform".
 - iii. W **Player Settings**:
 - iv. **Other Settings**:
 - v. **Minimum API Level**: Android 7.0 (API 24) lub wyższy.
 - vi. **Scripting Backend**: IL2CPP.
 - vii. **Target Architectures**: ARMv7 i ARM64.
- e. iOS:
 - i. **File > Build Settings**, wybierz "iOS".
 - ii. Kliknij "Switch Platform".
 - iii. W **Player Settings**:
 - iv. Włącz "Camera Usage Description" (np. "Aplikacja wymaga kamery do AR").
- f. Dodaj AR Session:
 - i. W hierarchii kliknij **Create > XR > AR Session**.
 - ii. Dodaj też **AR Session Origin** (**Create > XR > AR Session Origin**).

Rozpoznawanie płaszczyzn i spawnowanie obiektów przez dotyk

Krok 1: Przygotowanie sceny

1. W hierarchii upewnij się, że masz:
 - o **AR Session**.
 - o **AR Session Origin**.
2. Do **AR Session Origin** dodaj komponenty:
 - o **AR Plane Manager** (do wykrywania płaszczyzn, można ustawić **Plane Prefab**, by był widoczny po wykryciu).
 - o **AR Raycast Manager** (do rzucania promieni na płaszczyzny).
3. Stwórz prefabrykat obiektu do spawnowania:

Wykrywanie obrazka/wzoru i spawnowanie obiektu

Krok 1: Przygotowanie obrazu referencyjnego

1. Przygotuj obraz:
 - o Stwórz wyraźny obraz (np. PNG, 500x500 px) z unikalnym wzorem.
 - o Zapisz go w folderze projektu (np. "Assets/Images/Marker.png").
2. Utwórz bibliotekę obrazów:
 - o W Project kliknij **Create > XR > Reference Image Library**.
 - o Nazwij ją "ARImageLibrary".
 - o Dodaj obraz: kliknij "+", przeciągnij "Marker.png", ustaw "Physical Size" (np. 0.1 m x 0.1 m).

Krok 2: Konfiguracja sceny

1. Do **AR Session Origin** dodaj komponent **AR Tracked Image Manager**.
2. W Inspectorze:
 - o W polu "Serialized Library" przypisz "ARImageLibrary".
 - o W polu "Tracked Image Prefab" przypisz prefabrykat "SpawnObject" (ten sam co wcześniej).

Krok 3: Skrypt do śledzenia obrazu

1. Utwórz nowy skrypt "ARTrackImage".

```

2. using UnityEngine;
3. using UnityEngine.XR.ARFoundation;
4. using UnityEngine.XR.ARSubsystems;
5.
6. public class ARTrackImage : MonoBehaviour
7. {
8.     private ARTrackedImageManager trackedImageManager;
9.
10.    void Awake()
11.    {
12.        trackedImageManager = GetComponent<ARTrackedImageManager>();
13.    }
14.
15.    void OnEnable()
16.    {
17.        trackedImageManager.trackedImagesChanged += OnTrackedImagesChanged;
18.    }
19.
20.    void OnDisable()
21.    {
22.        trackedImageManager.trackedImagesChanged -= OnTrackedImagesChanged;
23.    }
24.
25.    void OnTrackedImagesChanged(ARTrackedImagesChangedEventArgs eventArgs)
26.    {
27.        foreach (var trackedImage in eventArgs.added)

```

Krok 2: Skrypt do spawnowania

1. Utwórz nowy skrypt:
 - o W Project kliknij **Create > C# Script**, nazwij go "ARPlaceObject".
2. Otwórz skrypt i wklej poniższy kod:

```

2. using UnityEngine;
3. using UnityEngine.XR.ARFoundation;
4. using UnityEngine.XR.ARSubsystems;
5. using System.Collections.Generic;
6.
7. public class ARPlaceObject : MonoBehaviour
8. {
9.     public GameObject objectToSpawn; // Prefabrykat do spawnowania
10.    private ARRaycastManager raycastManager;
11.    private List<ARRaycastHit> hits = new List<ARRaycastHit>();
12.
13.    void Start()
14.    {
15.        raycastManager = GetComponent<ARRaycastManager>();
16.    }
17.
18.    void Update()
19.    {
20.        if (Input.touchCount > 0 && Input.GetTouch(0).phase == TouchPhase.Began)
21.        {
22.            Vector2 touchPosition = Input.GetTouch(0).position;
23.
24.            if (raycastManager.Raycast(touchPosition, hits, TrackableType.Planes))
25.            {
26.                Pose hitPose = hits[0].pose;
27.                Instantiate(objectToSpawn, hitPose.position, hitPose.rotation);
28.            }
29.        }
30.    }
31. }

```

3. Podłącz skrypt:

- o Przeciągnij skrypt na **AR Session Origin**.
- o W Inspectorze, w polu "Object To Spawn", przypisz prefabrykat "SpawnObject".

```

28. {
29.     Debug.Log("Wykryto obraz: " + trackedImage.referenceImage.name);
30.     UpdateObjectPosition(trackedImage);
31. }
32.
33. foreach (var trackedImage in eventArgs.updated)
34. {
35.     UpdateObjectPosition(trackedImage);
36. }
37. }
38.
39. void UpdateObjectPosition(ARTrackedImage trackedImage)
40. {
41.     if (trackedImage.trackingState == TrackingState.Tracking)
42.     {
43.         trackedImage.transform.position = trackedImage.transform.position;
44.         trackedImage.gameObject.SetActive(true);
45.     }
46.     else
47.     {
48.         trackedImage.gameObject.SetActive(false);
49.     }
50. }
51. }

```

52. Podłącz skrypt do **AR Session Origin**.