1. Import mapbox sdk for unity
2. Assets🡪 mapbox🡪 examples🡪 prefab scenes🡪 locationBased
3. Make prefab of location provider and player game object inside ‘locationBasedGame’
4. Copy the prefabs to the previous projects’ scene
5. On map gameObject
   1. Uncheck initialize on start
   2. Add component 🡪 initialize map with location provider
      1. Open the script
      2. IlocationProvider gives the location of our phone using gps update
      3. Add map gameobject in the script
   3. In locationProvider gameobject
      1. Add map in the map manager
      2. In editor child:
         1. Set latitude and longitude of the place you want
6. **IN NEW MAPBOX:**
   1. Open new locationBased scene
   2. Create new gameobject inside locationProvider parent
      1. Add editorLocationProvider Script
      2. Click autofire event
      3. Accuracy – 100 (I put 100 myself)
      4. Latitude longitude of the place
   3. In locationProvider parent
      1. Drag and drop the new gameobject to the EditorLocationProvider field.

**COMBINE ARCORE AND MAPBOX:**

* + - 1. Import arcore sdk for unity
      2. Arcore🡪examples🡪helloar🡪scenes🡪helloar
         1. Drag and drop to the hierarchy
         2. Drag arcore device from hello ar to the mapbox scene
         3. Similarly, point cloud and environmental light, canvas, example controller drag and drop to mapbox scene
         4. Delete main camera
         5. Remove hello ar form the hierarchy
      3. Go to helloar script inside examplecontroller game object
         1. Delete the code inside if statement :

if (Frame.Raycast(touch.position.x, touch.position.y, raycastFilter, out hit))

**BUILD APP:**

1. Create a script named AnchorMap
2. using System.Collections;
3. using System.Collections.Generic;
4. using UnityEngine;
5. using GoogleARCore;
7. public class AnchorMap : MonoBehaviour {
9. // Update is called once per frame
10. void Update () {
11. //if player has not touched the screen, we are done with this update
12. Touch touch;
13. if(Input.touchCount<1||(touch = Input.GetTouch(0)).phase!=TouchPhase.Began)
14. {
15. return;
16. }
17. //raycast against the point where player touched
18. TrackableHit hit;
19. TrackableHitFlags raycastfilter = TrackableHitFlags.PlaneWithinBounds | TrackableHitFlags.PlaneWithinPolygon | TrackableHitFlags.FeaturePointWithSurfaceNormal;
20. if(Frame.Raycast(touch.position.x, touch.position.y, raycastfilter, out hit))
21. {
22. var anchor = hit.Trackable.CreateAnchor(hit.Pose );
23. transform.position = anchor.transform.position;
24. }
25. }
26. }
    1. Save the script and add it to the map gameobject
    2. Open initializeMapWith LocationProvider script :
       1. Inside void LocationProvider\_OnLocationUpdated(Unity.Location.Location location)
       2. Put transform.position = new Vector3(0, 5, 0);
    3. In unity editor:
       1. Scale map x,y,z to 0.005
       2. Scale player to 0.02
       3. Make player the child of map GameObject
          1. If map moves, player should also move
    4. In player settings:
       1. Other settings🡪 multithreaded rendering 🡪 false
       2. Give unique package name
       3. Minimum API ANDROID 7
       4. ARCORE supported in xr settings