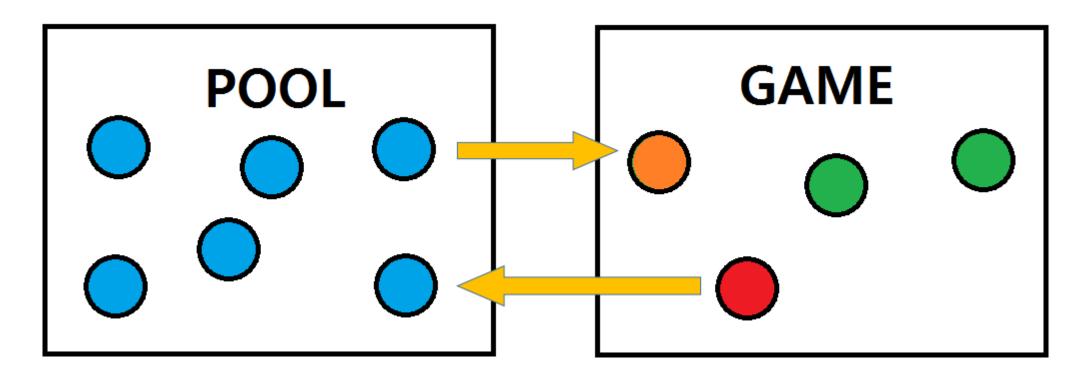
Unity -Pooling

NHN NEXT 서형석

- 메모리 풀 미리 구성된 메모리를 필요할 때 사용 후 돌려보내는 방식



- EnemyManager에 메모리 풀 적용

```
public class EnemyManager: MonoBehaviour
 List<GameObject> enemyPool = new List<GameObject>();
      poolSize = 10;
 void Start()
    for( int i=0; i<poolSize ; ++i )</pre>
        GameObject obj = Instantiate( enemy ) as GameObject;
        obj.name = "Enemy_" + i;
        obj.SetActive( false );
        enemyPool.Add(obj);
```

```
public class EnemyManager: MonoBehaviour
 void Update()
  if (deltaSpawnTime > spawnTime)
    deltaSpawnTime = 0.0f;
   // GameObject enemyObj = Instantiate(enemy) as GameObject;
     GameObject enemyObj = enemyPool.Find( item => item.activeSelf == false );
      if( enemyObj == null )
        return;
     enemyObj.SetActive( true );
     Vector3 spawnPos = playerTransform.forward * Random.Range(5.0f, 10.0f);
     spawnPos.x += Random.Range(-10.0f, 10.0f);
     spawnPos.z += Random.Range(0.0f, 5.0f);
     spawnPos.y = 0.1f;
     enemyObj.transform.position = spawnPos;
 } // end of if
```

```
public class Zombie: MonoBehaviour
 void OnEnable()
     InitZombie();
 void InitZombie()
    if(characterController != null)
        characterController.enabled = true;
     healthPoint = 5;
     enemyState = ENEMYSTATE.IDLE;
     PlayIdle();
IEnumerator DeadProcess()
     // Destroy( gameObject );
     gameObject.SetActive( false );
```

- 과제 : 본인이 생각하는 유연한 메모리 풀 매니저를 구성해 보자.