

언리얼 컨테이너 라이브러리 I – Array와 Set

(Unreal Container Library I – Array & Set)

강의 내용

언리얼에서 제공하는 대표 컨테이너 라이브러리의
동작 원리와 활용 방법을 예제를 통해 살펴보기

강의 목표

- 언리얼 대표 컨테이너 라이브러리 TArray, TSet의 내부 구조 이해
- 각 컨테이너 라이브러리의 장단점을 파악하고, 알맞게 활용하는 방법의 학습

언리얼 컨테이너 라이브러리

언리얼 컨테이너 라이브러리

- 언리얼 엔진이 자체 제작해 제공하는 자료구조 라이브러리
- 줄여서 UCL(Unreal Container Library)라고도 함.
- 언리얼 오브젝트를 안정적으로 지원하며 다수 오브젝트 처리에 유용하게 사용됨.
- 언리얼 C++은 다양한 자료구조 라이브러리를 직접 만들어 제공하고 있음.
- 실제 게임 제작에 유용하게 사용되는 라이브러리로 세 가지를 추천함.

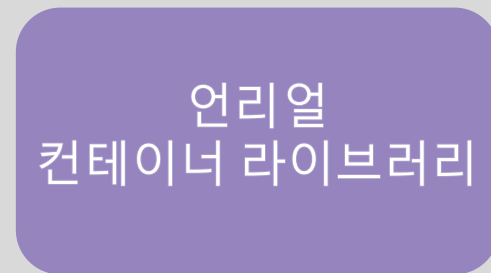
TArray, TArrayBuilder, TArrayView, TBasicArray
FBinaryHeap, TBitArray, TChunkedArray
TCircularBuffer, TCircularQueue, TClosableMpScQueue, TDeque
TEnumAsByte, FHashTable, TIndirectArray
TIntrusiveDoubleLinkedListNode, TLinkedList
TMap, TMapBuilder, TQueue, TSet, TSortedMap, TStaticArray
TMpScQueue, TSparseArray, TScriptSparseArray, TStringView
TTripleBuffer, TUnion, TTransArray, TRingBuffer, TMRUArray



TArray, TMap, TSet

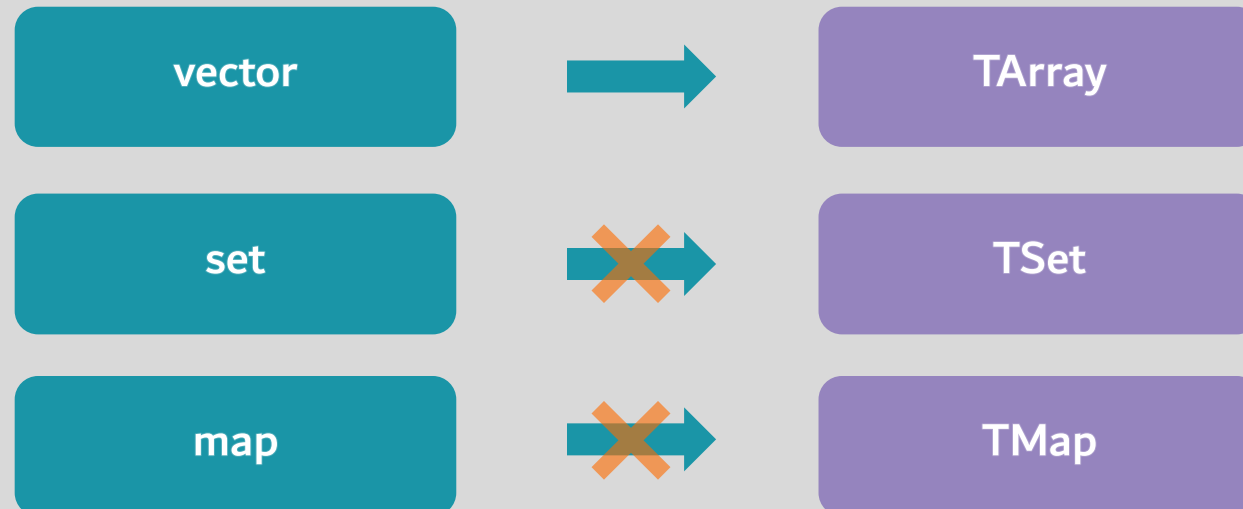
C++ STL과 언리얼 컨테이너 라이브러리의 차이점

- C++ STL은 범용적으로 설계되어 있다.
- C++ STL은 표준이기 때문에 호환성이 높다.
- C++ STL에는 많은 기능이 얹여 있어 컴파일 시간이 오래 걸림.
- 언리얼 컨테이너 라이브러리는 언리얼 엔진에 특화되어 있음.
- 언리얼 컨테이너 라이브러리는 언리얼 오브젝트 구조를 안정적으로 지원한다.
- 언리얼 컨테이너 라이브러리는 가볍고 게임 제작에 최적화되어 있음.



언리얼 C++ 주요 컨테이너 라이브러리

- 두 라이브러리의 이름과 용도는 유사하지만, 내부적으로 다르게 구현되어 있음.
 - TArray : 오브젝트를 순서대로 담아 효율적으로 관리하는 용도로 사용
 - TSet : 중복되지 않는 요소로 구성된 집합을 만드는 용도로 사용
 - TMap : 키, 밸류 조합의 레코드를 관리하는 용도로 사용

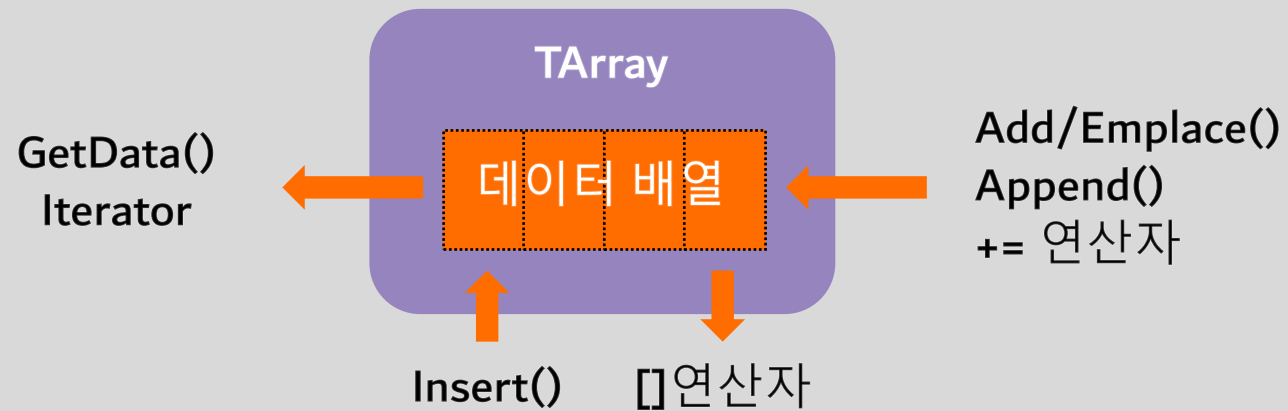


TArray의 구조와 활용

TArray 개요

- TArray는 가변 배열(Dynamic Array) 자료구조
- STL의 vector와 동작 원리가 유사함.
- 게임 제작에서는 가변 배열 자료구조를 효과적으로 활용하는 것이 좋음.
 - 데이터가 순차적으로 모여있기 때문에 메모리를 효과적으로 사용할 수 있고 캐시 효율이 높다.
 - 컴퓨터 사양이 좋아지면서, 캐시 지역성(Locality)으로 인한 성능 향상은 굉장히 중요해짐.
 - 임의 데이터의 접근이 빠르고, 고속으로 요소를 순회하는 것이 가능.
- 가변 배열의 단점
 - 맨 끝에 데이터를 추가하는 것은 가볍지만, 중간에 요소를 추가하거나 삭제하는 작업은 비용이 큼
- 데이터가 많아질 수록 검색, 삭제, 수정 작업이 느려지기 때문에, 많은 수의 데이터에서 검색 작업이 빈번하게 일어난다면 TArray대신 TSet을 사용하는 것이 좋음.

TArray의 내부 구조



TArray



<https://bit.ly/uarraykr>

<https://docs.unrealengine.com/5.1/ko/array-containers-in-unreal-engine/>

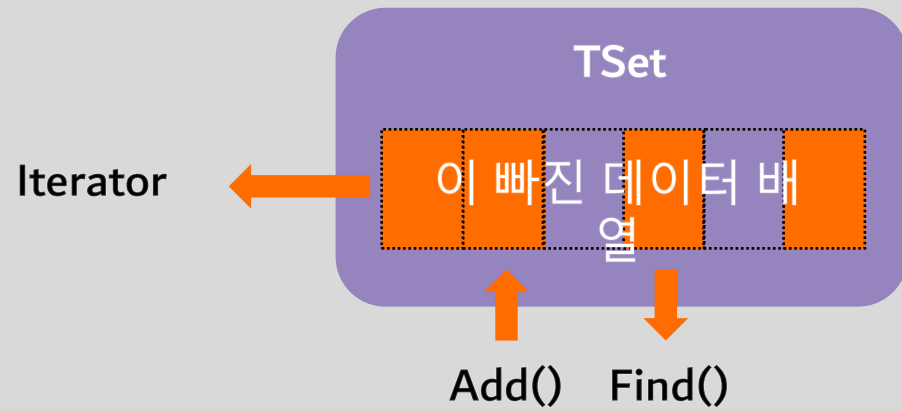
TArray의 활용1

TSet의 구조와 활용

TSet의 특징

- STL의 set과 언리얼 TSet의 비교
 - STL set의 특징
 - STL set은 이진 트리로 구성되어 있어 정렬을 지원함.
 - STL set은 메모리 구성이 효율적이지 않음.
 - STL set은 요소가 삭제될 때 균형을 위한 재구축이 일어날 수 있음.
 - STL set의 모든 자료를 순회하는데 적합하지 않음.
 - 언리얼 TSet 특징
 - TSet은 해시테이블 형태로 키 데이터가 구축되어 있어 빠른 검색이 가능함.
 - TSet은 동적 배열의 형태로 데이터가 모여있음
 - TSet의 데이터는 빠르게 순회할 수 있음.
 - TSet의 데이터는 삭제해도 재구축이 일어나지 않음.
 - TSet의 자료에는 비어있는 데이터가 있을 수 있음.
- 따라서 STL set과 언리얼 TSet의 활용 방법은 서로 다르기 때문에 주의할 것.
- STL의 unordered_set과 유사하게 동작하지만 동일하진 않음.
- TSet은 중복 없는 데이터 집합을 구축하는데 유용하게 사용할 수 있음.

TSet의 내부 구조



TSet



<https://bit.ly/uetssetkr>

<https://docs.unrealengine.com/5.1/ko/set-containers-in-unreal-engine/>

TSet의 활용1 – TArray와 비교

자료구조의 시간 복잡도 비교

- 각 자료구조의 시간복잡도(Time Complexity)

	TArray	TSet
접근	$O(1)$	$O(1)$
검색	$O(N)$	$O(1)$
삽입	$O(N)$	$O(1)$
삭제	$O(N)$	$O(1)$

빈틈없는 메모
가장 높은 접근성
가장 높은 순회성

빠른 중복 감지

정리

언리얼 컨테이너 라이브러리

1. TArray, TSet 컨테이너 라이브러리 내부 구조와 활용 방법
2. 디버그 빌드를 사용해 메모리 정보를 확인하는 방법의 학습
3. 두 컨테이너 라이브러리가 가진 특징의 이해