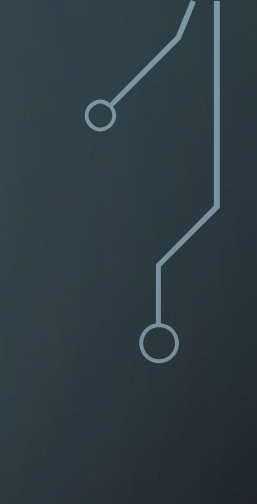# OUTLINE

1. TArray Function Blueprint Comparison

2. TArray Iteration Types

3. TArray Important Functions

TARRAY BLUEPRINT/C++ FUNCTION COMPARISON

- Add (Blueprint)   = Array.Add(element);

- Insert (Blueprint)   = Array.Insert(element,index);

- Append Array (Blueprint)   = Array.Append(array);

- Length (Blueprint)   = Array.Num();

- LastIndex (Blueprint)   = Array.Num() – 1;

- Get (Blueprint)   = Array[index];

TARRAY BLUEPRINT/C++ FUNCTION COMPARISON

- Contains Item (Blueprint)   = Array.Contains(element);

- Find Item (Blueprint)   = Array.Find(element) ;

  Array.Find(element,OutIndex);

- Remove Item (Blueprint)   = Array.Remove(element);

- Remove Index (Blueprint)   = Array.RemoveAt(index);

- Clear (Blueprint)   = Array.Empty();

TARRAY STANDARD FOR LOOP

```
TArray<FVector> Array;

….

Int32 Num = Array.Num();

for(int32 i=0; i < Num; i++)

{

        FVector& Vec = Array[i];

}
```

TARRAY RANGE BASE FOR EACH LOOP

```cpp
TArray<FVector> Array;

….

for(FVector& Val : Array)

{

    …

}

for(auto& Val : Array)

{

    …

}
```

TARRAY ITERATORS

```cpp
TArray<FVector> Array;

….

for (auto It = Array.CreateIterator(); It; ++It)

{

    FVector& Vec = *It;

}

for (auto It = Array.CreateConstIterator(); It; ++It)

{

    const FVector& Vec = *It;

}
```

TARRAY IMPORTANT FUNCTIONS RECAP

1. Add – Adds a new Element

2. Insert – Adds a new Element at a specific Index

3. Contains – Checks to see if array contains the element

4. Find – Finds the Index of the element, Returns INDEX_NONE if not found

5. Num – returns the number of elements

6. Empty – Clears the whole array

7. Remove – Removes an element from the array

8. RemoveAt – Removes the element at a given index