

시스템 관련 용어 I

2021/5/15

류한진

Singleton

- 소프트웨어 디자인패턴의 일종
- 필요한 순간
 - 클래스당 하나의 인스턴스만을 가져야 할 때
 - 하나의 인스턴스가 쉽게 접근되도록 해야 할 때
 - 클래스가 객체 할당에 있어 엄격해야 할 때
 - 클래스의 인스턴스 개수 제한을 뒤야 할 때
 - 전역변수 접근이 필요할 때
- 적용 방법
 - 생성자를 숨긴다
 - 하나의 인스턴스를 리턴하는 정적메소드를 만든다
- 사용 예시
 - 다른 디자인 패턴들의 기반으로 사용됨
 - 후술할 상태기계를 구현할 때 사용됨
 - 전역변수의 대체제로 사용됨
 - 전역 네임스페이스에 사소한 변수명을 선언하지 않아도 됨
 - 프로그램이 실행됨과 동시에 자원을 사용하는 전역변수와는 달리 선언되지 않으면 자원을 사용하지 않음

State Machine

- 동일한 입력이 들어왔더라도 상태에 따라 다른 출력을 내는 오토마타
 - 주로 유한상태기계를 칭한다
- 유한상태기계의 일반적인 구성요소는 다음과 같다;
 - 입력 : 상태함수와 출력함수로 처리될 수 있는 정의역
 - 출력 : 출력함수의 치역
 - 상태 : 상태함수와 출력함수의 결과에 영향을 미치는 값으로 상태기계는 이 값을 입력이 들어오기 전까지 동일하게 유지한다.
 - 상태함수 : 현재 상태와 입력에 따라 전이될 다음 상태를 결정하는 규칙
 - 출력함수 : 현재 상태와 입력에 따른 출력을 결정하는 규칙
 - 시작상태 : 상태기계에서 첫번째 입력이 들어오기 전까지 유지되는 상태
 - * 종료상태 : 해당 상태에 이르면 상태기계를 종료한다.
- 컴퓨터공학에서는 대표적으로 구문분석, 어휘분석과 인공지능에 사용되고 있다.

Unity Coroutine

- 코루틴은 분기를 일시정지해가며 분할처리되는 함수다.
 - 코루틴은 멀티스레딩의 장점을 갖지만 운영체제의 멀티태스킹과 더 유사하다.
 - 호출 후 메인 분기에서는 코루틴을 호출하는 코드를 작성 할 필요가 없다.
 - 코루틴 함수의 모든 진행상황과 메모리 리소스는 저장되었다 분기가 돌아왔을 때 재사용된다.
- Yield
 - Yield 키워드는 분기의 일시정지점이다.
 - 이를 선언 함으로서 해당 함수가 코루틴 함수임을 나타낸다.
 - 유니티 스크립트에서는 StartCoroutine()없이 실행만 해도 코루틴으로 실행된다.
- StartCoroutine()
 - 유니티 스크립트가 아닌 Unity C# 에서는 이 함수를 통해 코루틴 함수를 호출해야 한다.
- WaitForSeconds()
 - Yield 선언의 끝에 이 함수를 붙여 정해진 시간만큼의 딜레이를 가할 수 있다.
- 코루틴은 다음 요인들로 인해 종료된다.
 - 코루틴의 완료, StopCoroutine(), StopAllCoroutine(), SetActive(false), Destroy()