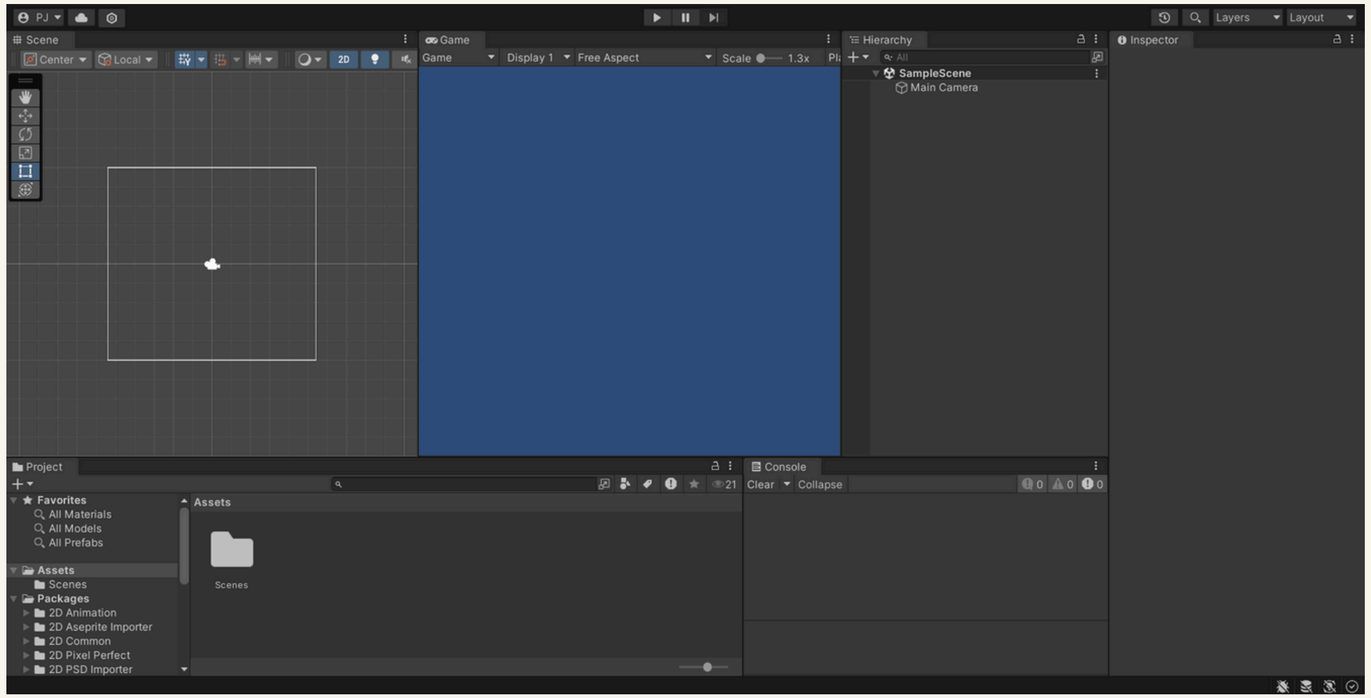
유니티 및 C# 스터디

made by 박준영

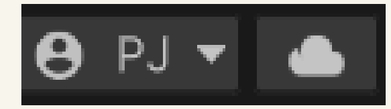
목차

유니티 인터페이스

■Toolbar

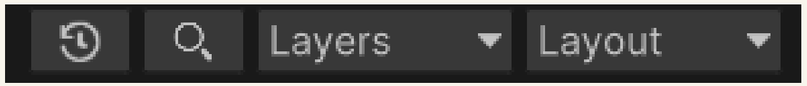


에디터 전체 화면



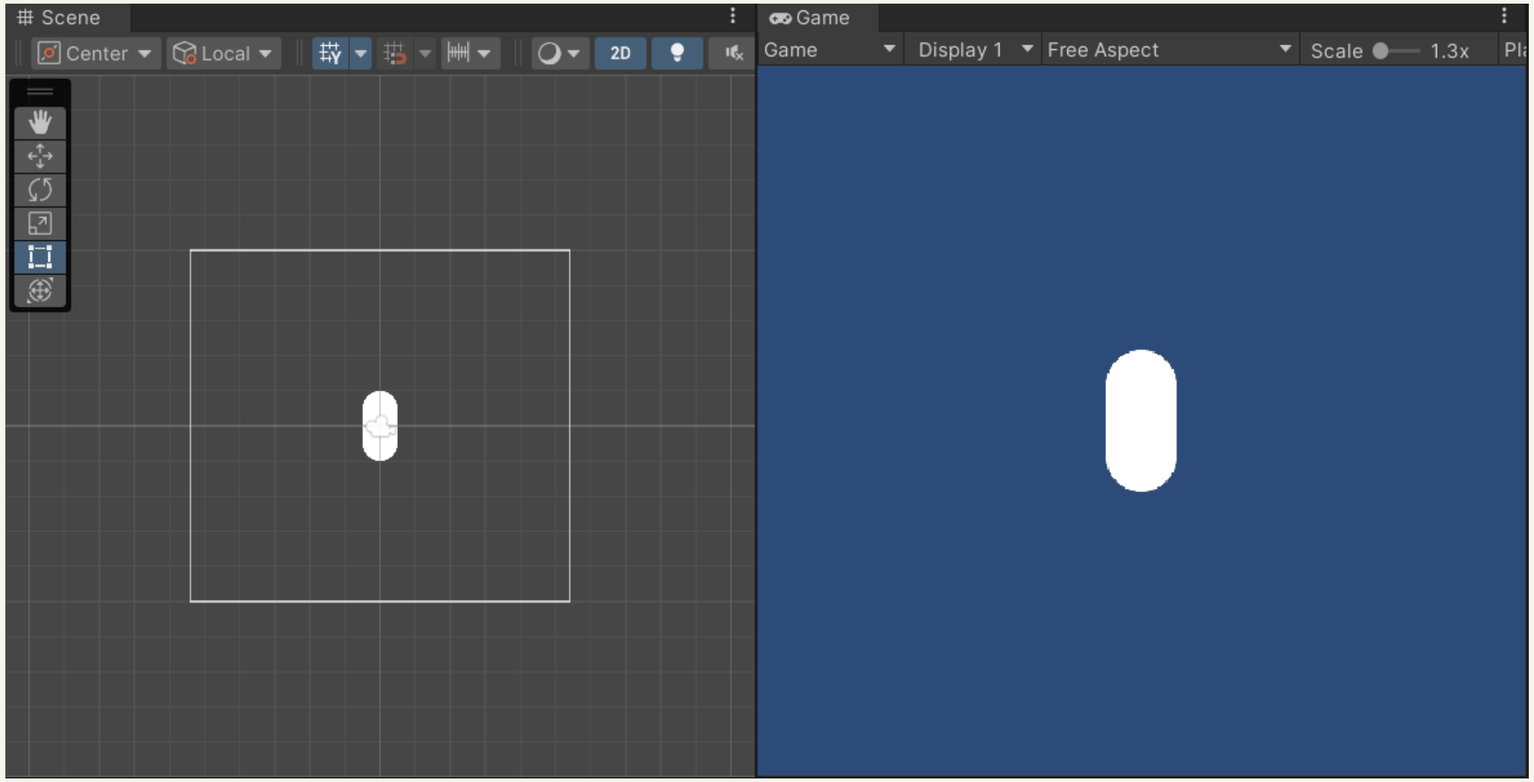
- 유니티 계정 엑세스
- 유니티 서비스 창 열기

- 게임 뷰에서 Play, Pause 및 Step 버튼 사용



- 유니티에서 수행 한 작업 확인, 취소, 다시시작 가능
- 검색창 열기
- 씬 뷰에 표시할 오브젝트를 관리
- 뷰 정렬을 변경, 새 레이아웃 저장 기존 레이아웃 로드

■Scene, Game



Scene, Game 창

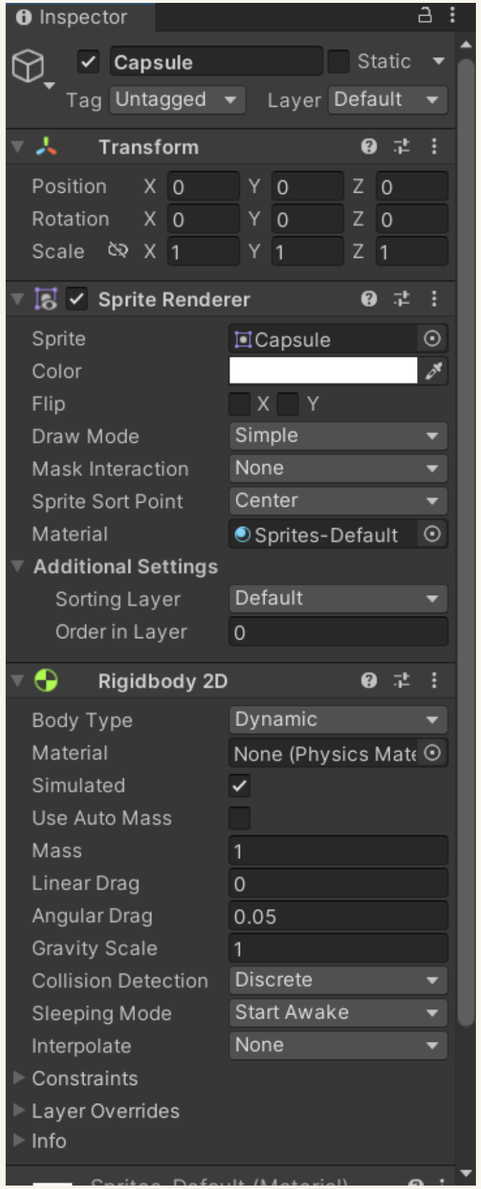
Scene

- 영화 세트장 역할
- 여러가지 물픔 배치
- 오브젝트 크기 조정 가능

Game

- 영화 카메라 역할
- 실제 게임 화면을 보여줌
- 화면 비율 조정 가능
- 시뮬레이터를 통해 다양한 기기에서 게임이 어떻게 보이는지 확인 가능

유니티 인터페이스



인스펙터 창

■Inspector

-오브젝트의 속성을 나타냄

-오브젝트의 컴포넌트를 관리할 수 있음

-비활성화 시 오브젝트의컴포넌트가 없는 것으로 인식

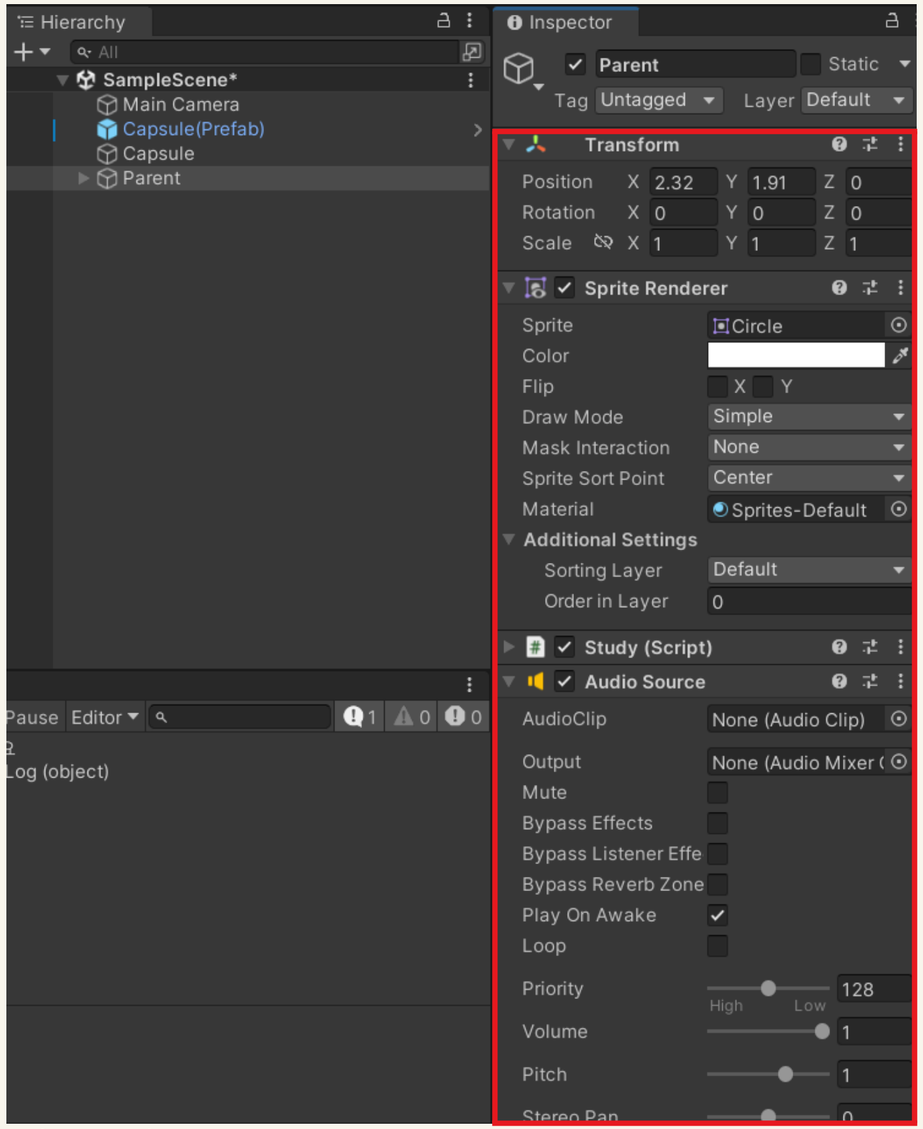
-Tag를 설정하여 오브젝트를 식별할 수 있게 함

-오브젝트가 플레이 중 움직일 수 있는지 체크

-오브젝트의 연산을 미리 수행하여 런타임 연산을 줄임

-렌더링 설정, 충돌 여부 설정, 레이케스팅 충돌을 담당

유니티 인터페이스



컴포넌트

Component

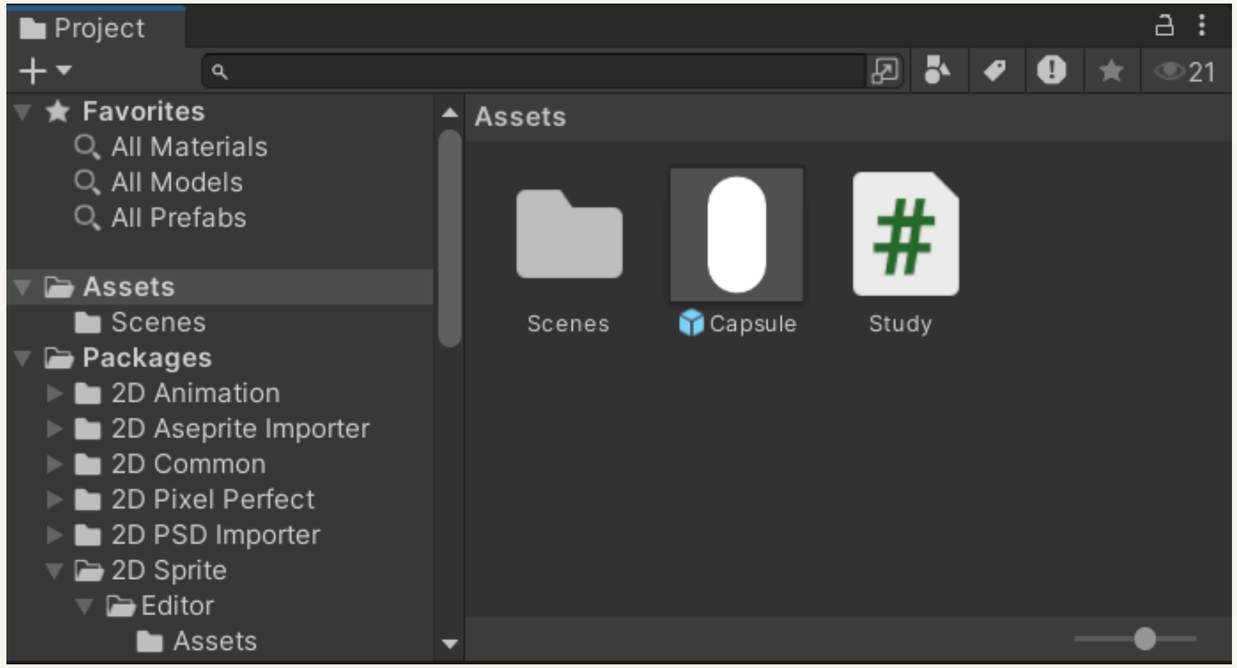
-게임 오브젝트에 부착할 수 있는 C# 스크립트 파일을 지칭

-게임 오브젝트에 부착하여 여러 기능을 부여함

-인스펙터 창에 게임 오브젝트가 가지고 있는 컴포넌트 목록이 표시

-Transform 컴포넌트로 위치를 정하고 Sprite Renderer 컴포넌트로 이미지를 나타냄

유니티 인터페이스



프로젝트 창

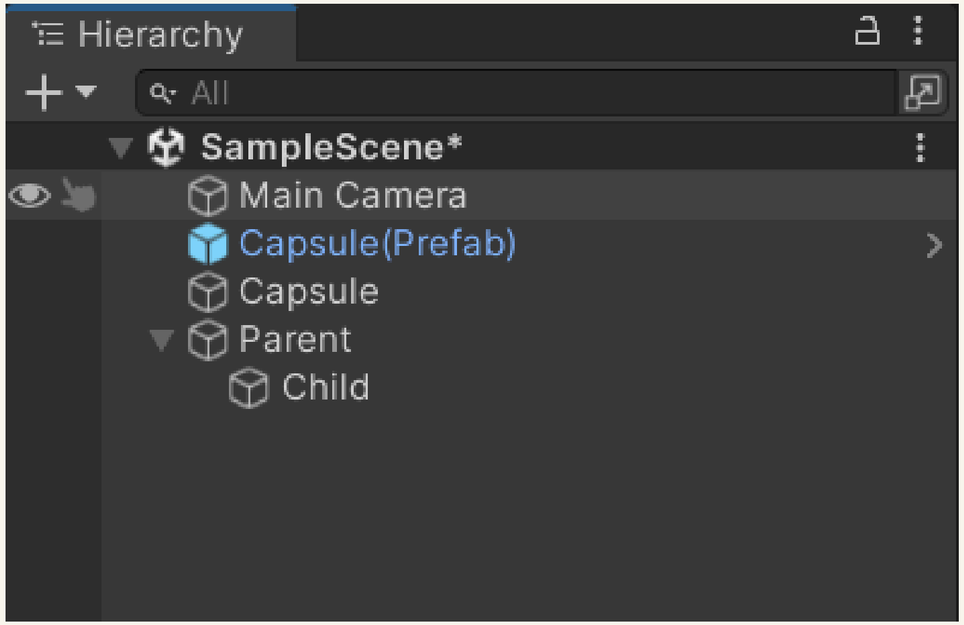
■Project

-소품 창고 역할

-씬, 이미지, 애니메이션, 사운드,
프리팹 등 여러 리소스 저장

-게임의 기능을 담고 있는 스크립트 저장

유니티 인터페이스

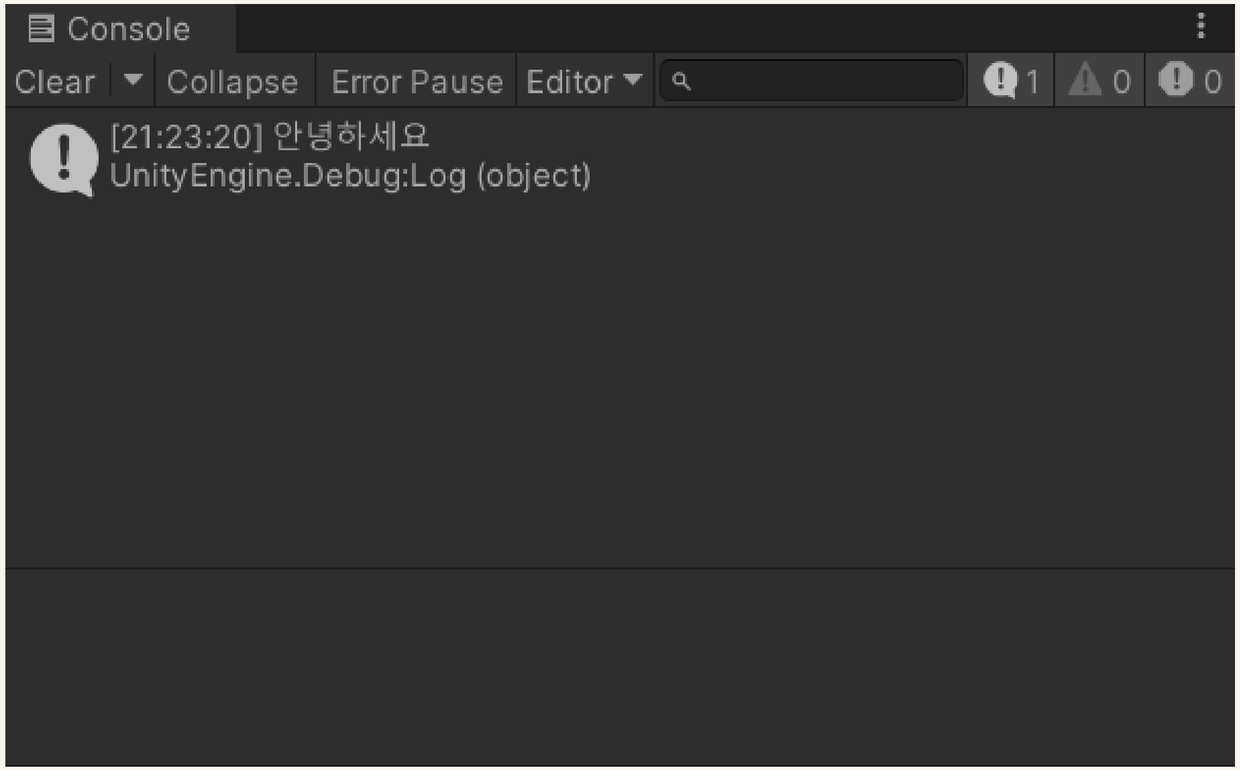


하이라키 창

■Hierarchy

-영화 세트장에 배치된 소품 리스트
-프로젝트 창에서 드래그 앤 드롭으로
컴포넌트를 입힐 수 있음
-오브젝트 간 계층구조 확인 가능
-프리팹 오브젝트는 파란색 표시

유니티 인터페이스



콘솔 창

■Console

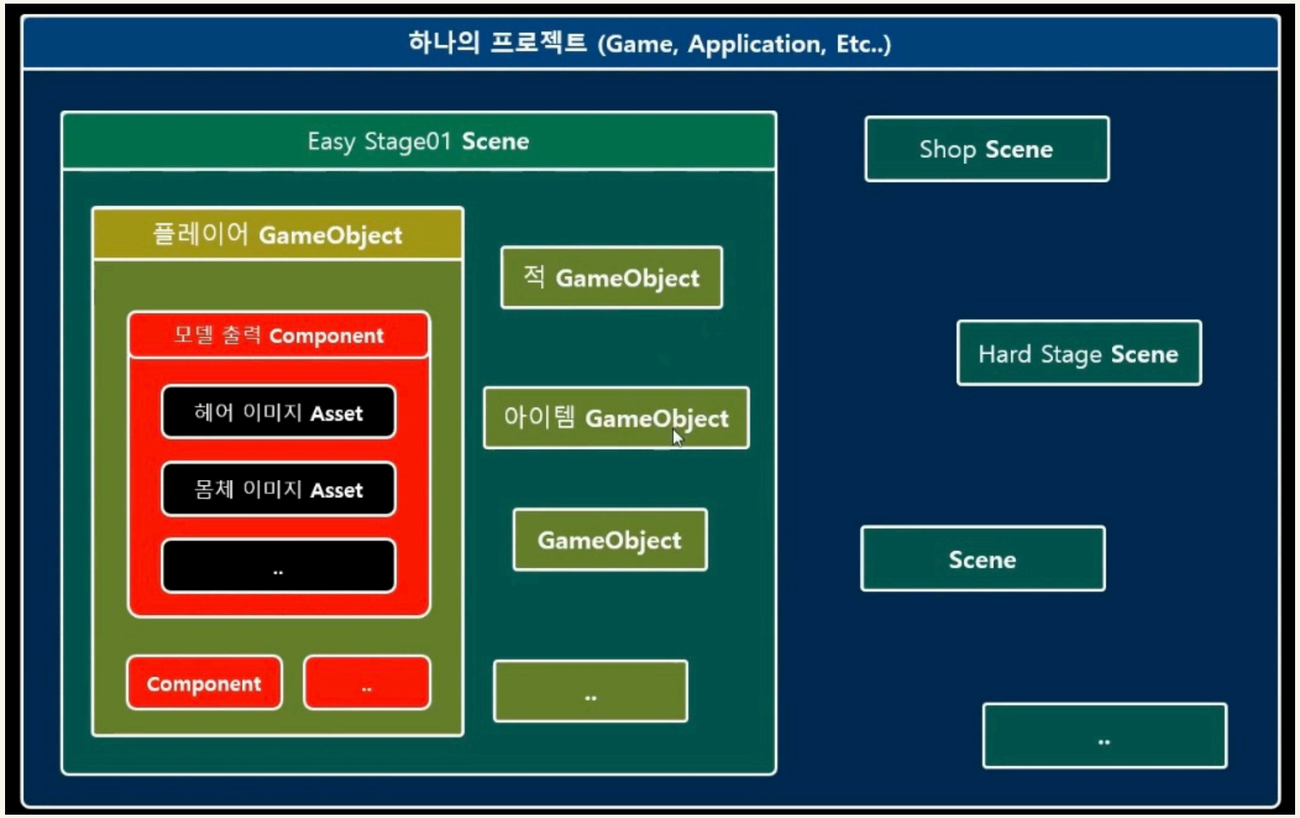
작성한 소스 코드에서 발생한
경고, 에러, 정보 출력

Clear : 콘솔 뷰에 출력된 모든 데이터 삭제

Collapse : 완전히 동일한 데이터를
묶어서 표기

Error Pause : 에디터에서 게임 실행 중
에러가 발생하면 프로그램을 “Pause” 시킴

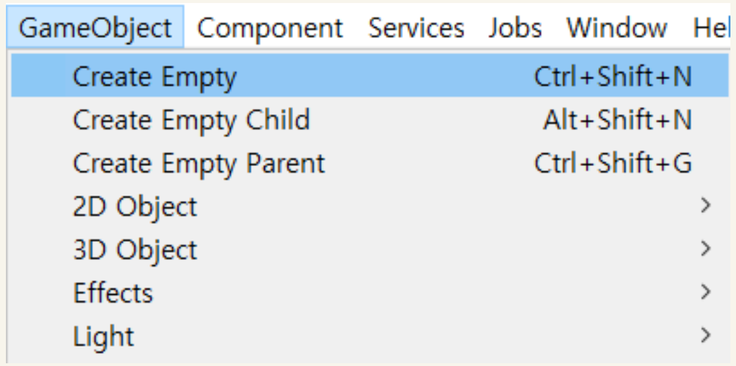
게임 오브젝트 종류

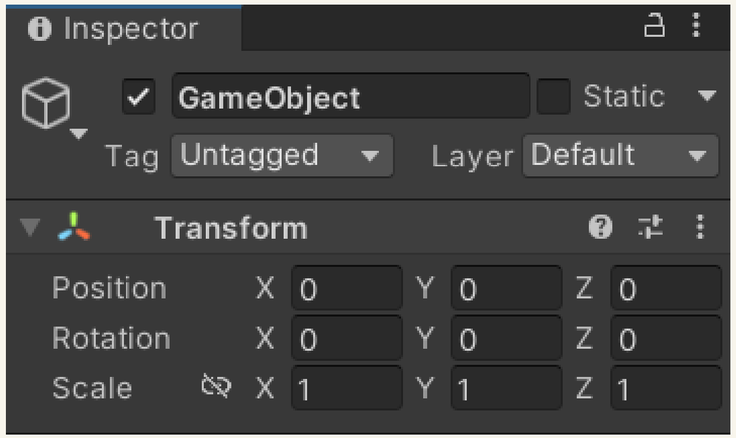


관계도

-하나의 프로젝트에는 여러 씬이 존재
-각 오브젝트는 다양한 컴포넌트를
가지고 있음
-여러 에셋을 이용하여 컴포넌트를 구성 후
최종적으로 게임 오브젝트를 완성

게임 오브젝트 종류





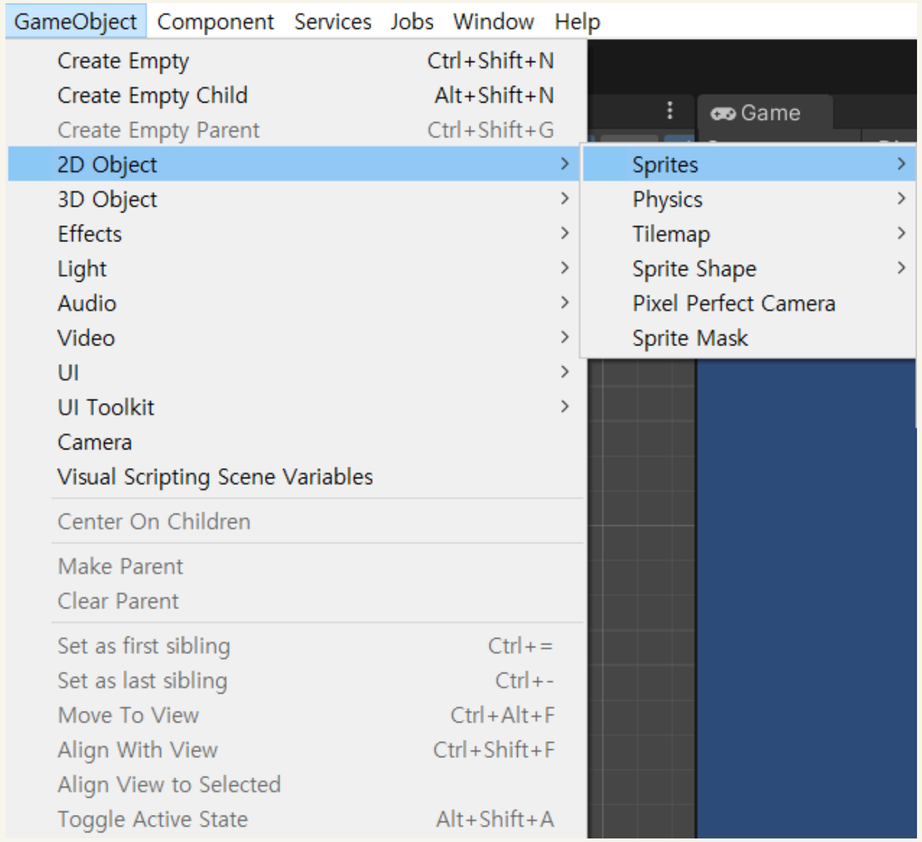
빈 오브젝트 생성법

■Empty Object

- Transform 컴포넌트만 붙어있는 오브젝트
- GameObject->Create Empty를 통해 생성 가능

- 게임에 보이지 않은 오브젝트이므로
  게임의 여러 요소를 처리하는
  컴포넌트를 추가해서 사용함
- ex) GameManager, StageController

게임 오브젝트 종류



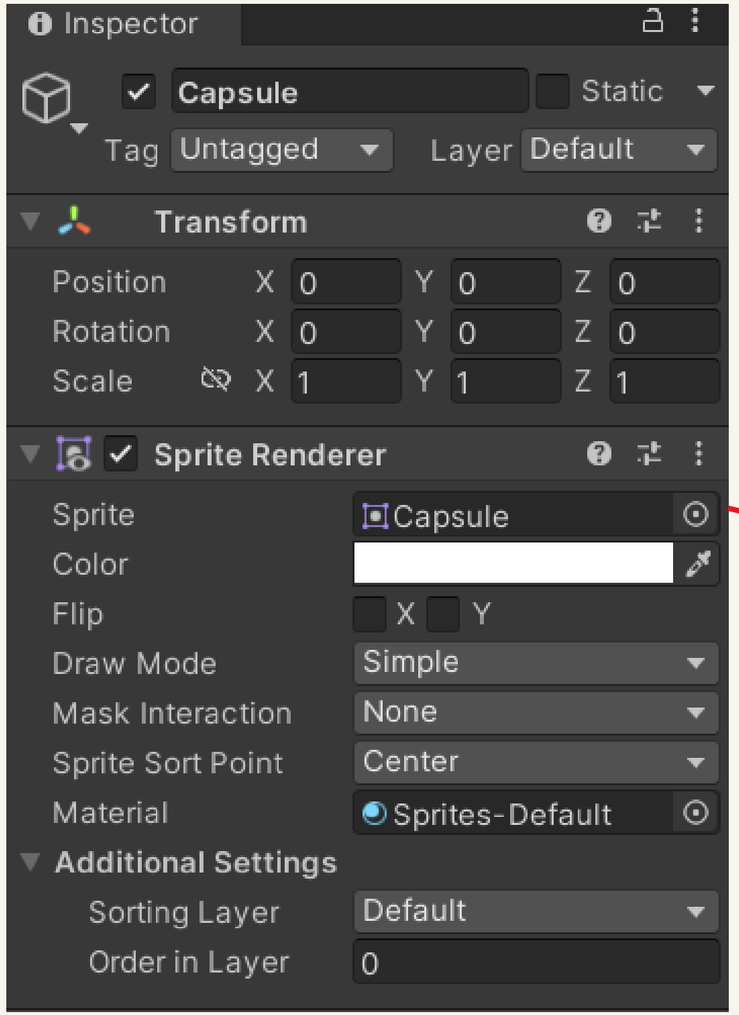
2D 오브젝트 생성법

■2D Object

- 게임에 배치할 수 있는 2D 오브젝트
- GameObject->2D Object를 통해 생성 가능

- Sprites : 게임 화면 2D 이미지를 보이게 하는 게임 오브젝트
- Physics : Sprites에 Collider와 RigidBody 컴포넌트를 추가한 오브젝트
- Tilemap : 타일 형태의 2차원 맵을 제작할 때 사용함
- Sprite Mask : Sprite 오브젝트를 숨기거나 보여주는데 사용

게임 오브젝트 종류



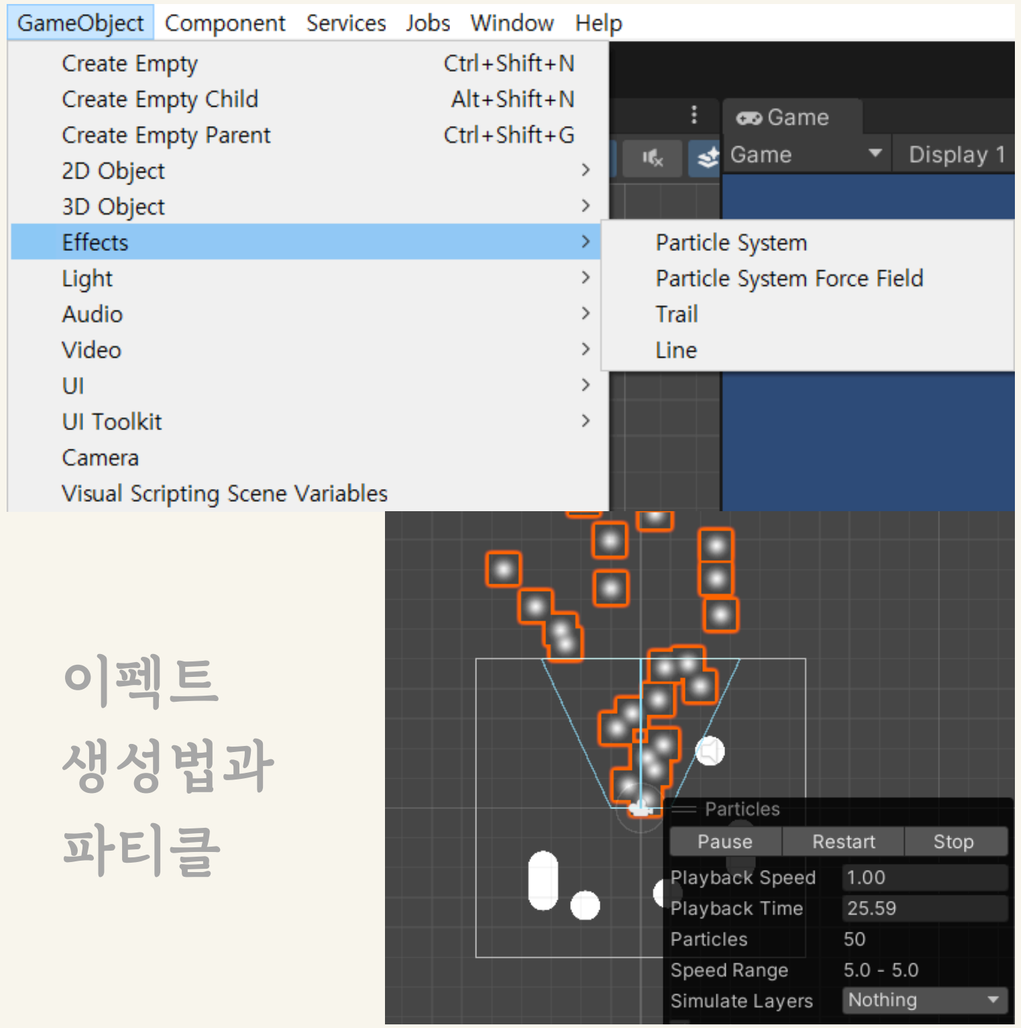
스프라이트 렌더러

Sprite

- 게임 화면에 2D 이미지를 보이게 하는 게임 오브젝트

- Sprite 변수에 적용된 에셋을 화면에 출력

게임 오브젝트 종류



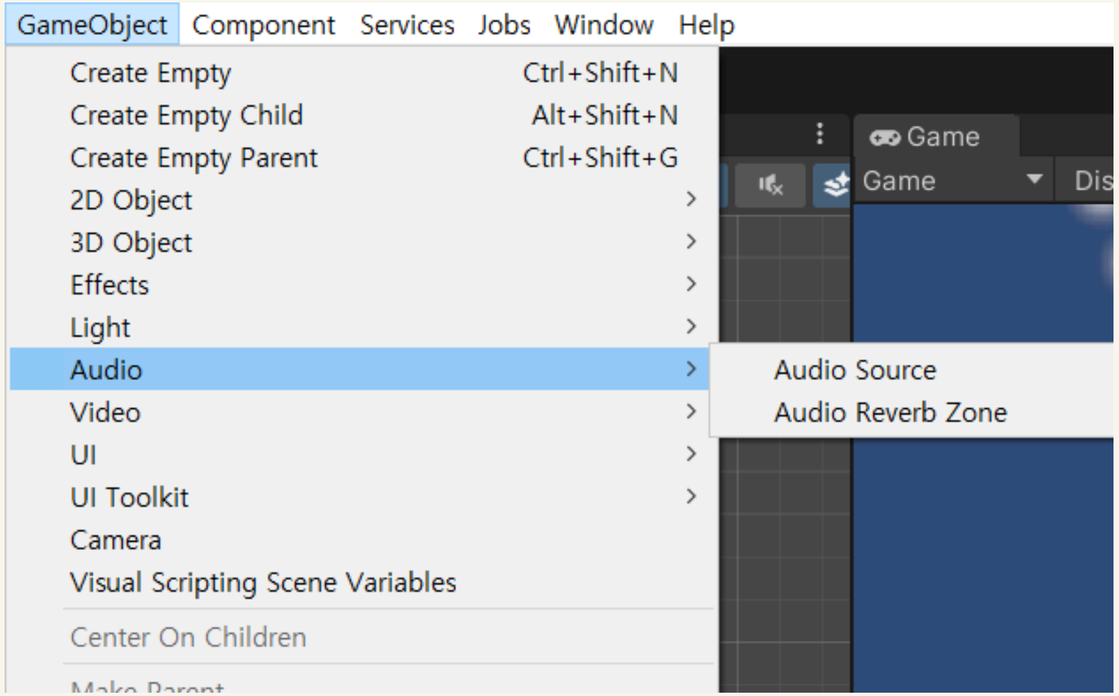
이펙트 생성법과 파티클

■Effects

- 여러 효과, 잔상, 선 그리기와 같은 오브젝트
- GameObject->Effects를 통해 생성 가능

- Particle System : 여러 효과를 제공하는 오브젝트
- Trail : 잔상 효과를 제공하는 오브젝트
- Line : 두 개 이상의 지점 배열을 가져와 각각의 점 사이에 직선을 그리는 오브젝트

게임 오브젝트 종류



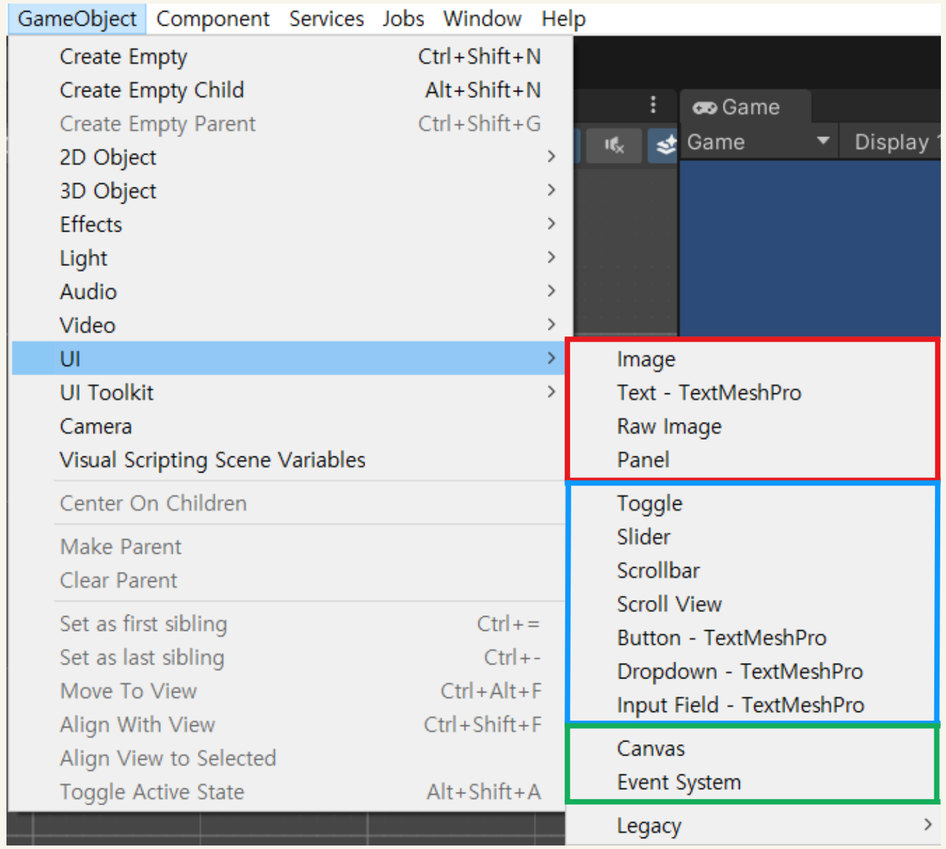
오디오 생성법

■Audio

- 게임 내에서 재생되는 사운드 오브젝트
- GameObject->Audio를 통해 생성 가능

- Audio Source : 소리를 내는 오브젝트
- Audio Reverb Zone : 동굴과 같은 실내의 울리는 잔향을 만들 때 사용
- 소리를 듣는 컴포넌트인 Audio Listener 컴포넌트는 카메라 오브젝트에 부착되어 있음

게임 오브젝트 종류



UI 생성법

■UI

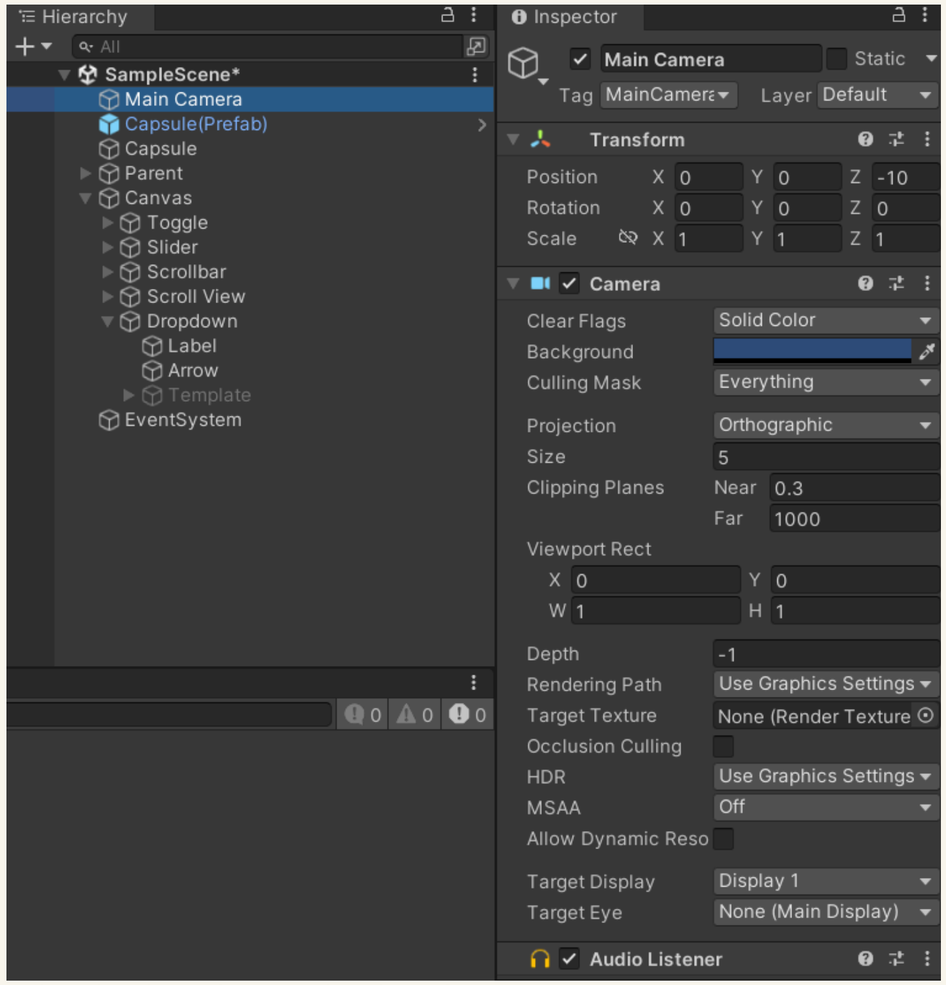
- 사용자가 게임과 상호작용 할 수 있는 GUI 오브젝트
- GameObject->UI를 통해 생성 가능

-텍스트와 이미지 오브젝트는 상호작용이 불가능

-버튼, 토글, 드롭다운과 같이상호작용이 가능한 오브젝트

-UI를 화면에 표시하고 상호작용을 가능하게 하는 오브젝트

게임 오브젝트 종류



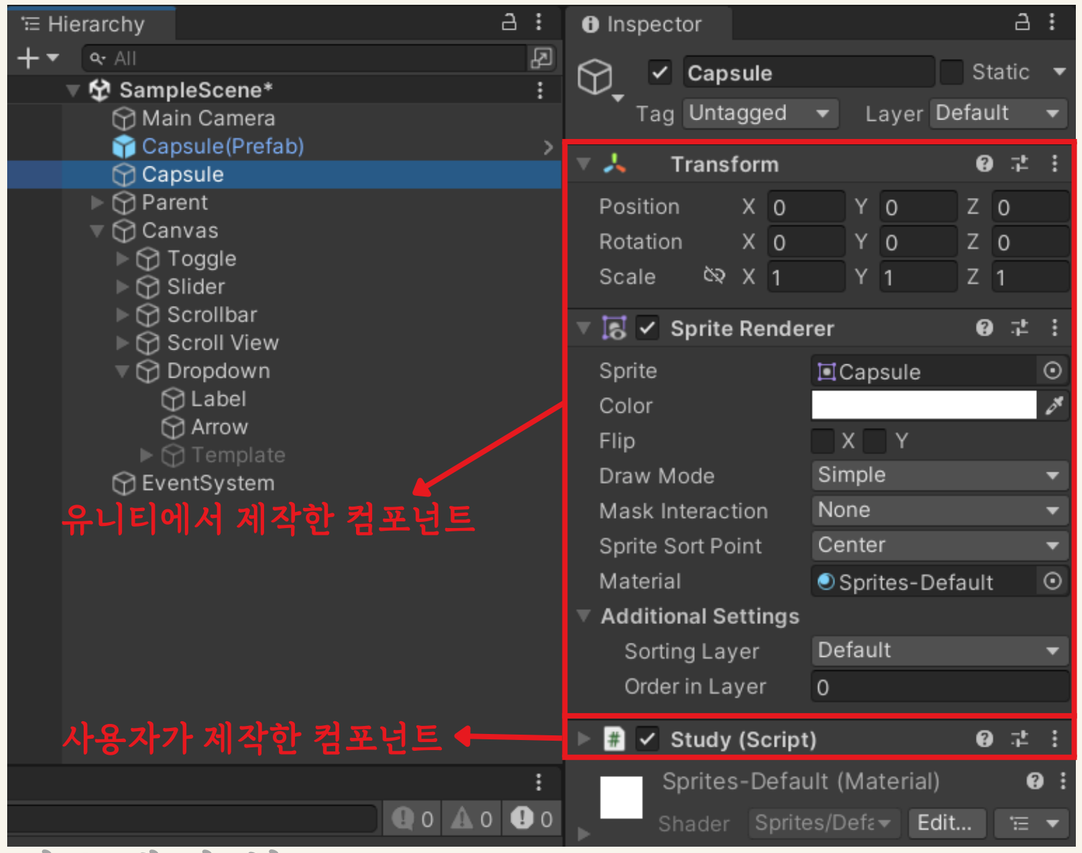
카메라 인스펙터 창

■Camera

- 사용자가 게임 화면을 볼 수 있게 해주는 컴포넌트
- 카메라 오브젝트는 씬에 최소 1개 이상 존재해야 함
- 여러 개의 카메라를 사용해 CCTV와 같은 연출 가능

Clear Flags : 빈 배경을 어떻게 채울지 결정하는 요소
-2D는 Solid Color, 3D는 Skybox를 주로 사용
Projection : 카메라 시점을 나타내며 2D,3D 시점이 존재함
Clipping Planes : 카메라가 볼 수 있는 시야 거리
-Near은 시작 지점을 나타내며 Far은 끝 지점을 나타냄
Viewport Rect : 카메라가 화면에 출력하는 영역을 설정

이벤트 함수

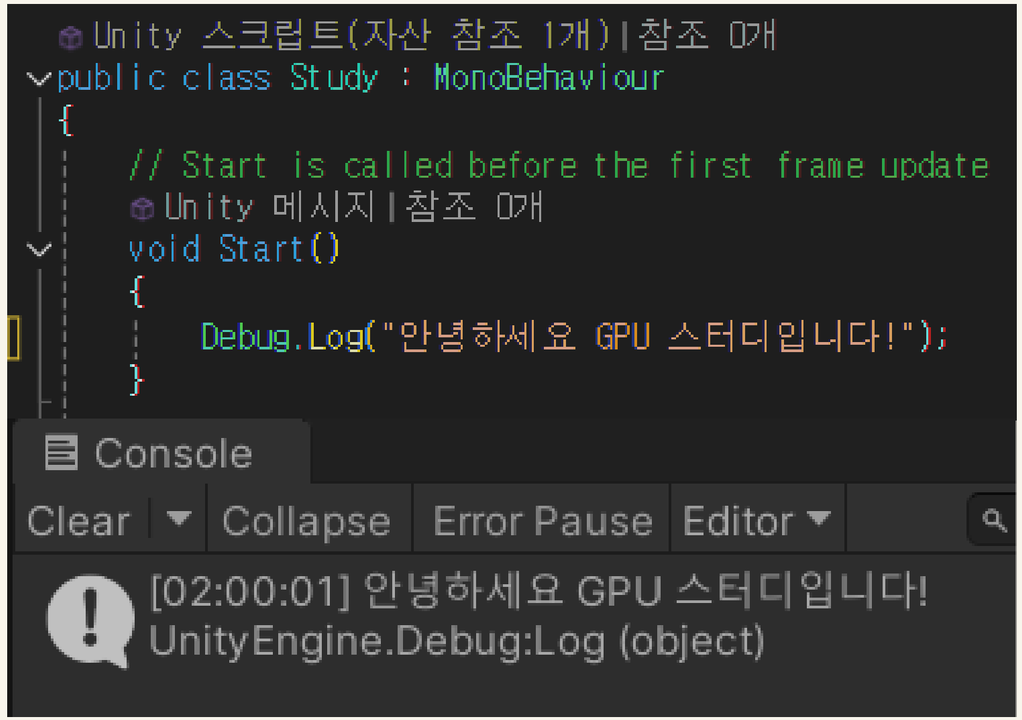


인스펙터 창

■Script

- 게임 오브젝트에 주어지는 각종 명령 제어
- 게임에 사용되는 여러 오브젝트 생성, 삭제 및 관리
- 게임을 관리하는 게임 내 시스템 구현
- 사용자가 정의한 스크립트는 컴포넌트와 같기 때문에 스크립트 내부에서 다른 컴포넌트에 접근하여 정보를 얻거나 수정이 가능함

이벤트 함수



스크립트와 결과

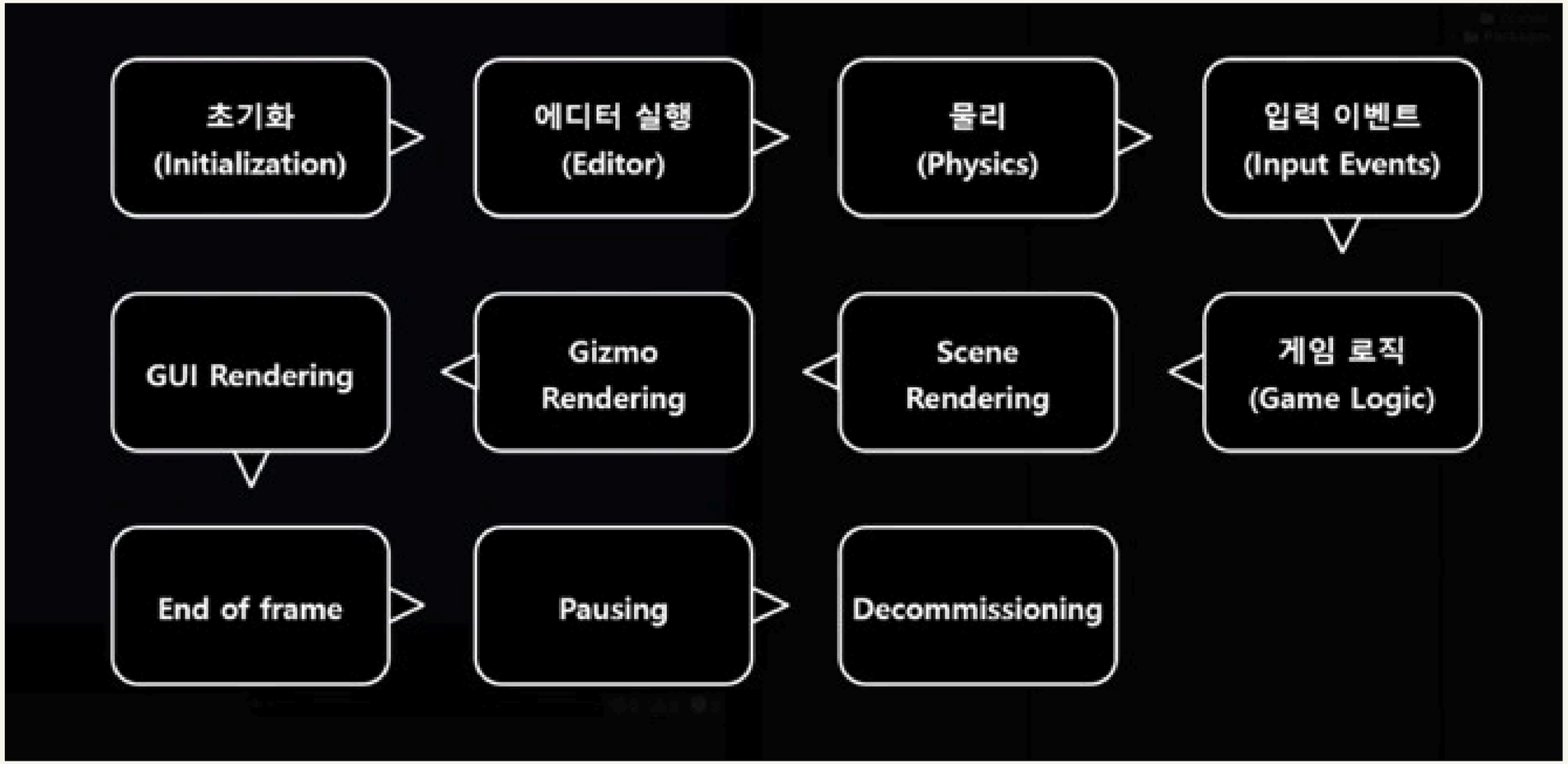
스크립트를 컴포넌트로 적용하기 위한 조건

1. 스크립트 파일 이름과 클래스 이름이 같아야 함
2. 부모 클래스로 MonoBehaviour를 상속받아야 함

콘솔뷰에 데이터 출력하는 방법

- Debug.Log(“내용”), print(“내용”)을 통해 출력 가능

이벤트 함수

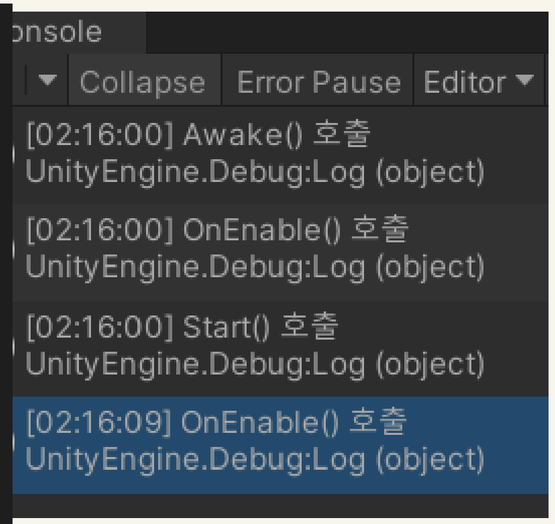

초기화
(Initialization)
에디터 실행
(Editor)
물리
(Physics)
입력 이벤트
(Input Events)
게임 로직
(Game Logic)
Scene
Rendering
Gizmo
Rendering
GUI Rendering
End of frame
Pausing
Decommissioning

호출 순서

이벤트 함수

```
Unity 메시지 | 참조 0개
void Awake()
{
    Debug.Log("Awake() 호출");
}
참조 0개
void Start()
{
    Debug.Log("Start() 호출");
}

private void OnEnable()
{
    Debug.Log("OnEnable() 호출");
}
```



초기화 함수 스크립트와 결과

- 호출순서 Awake()->OnEnable()->Start()

■Awake()

- 데이터를 초기화하는 목적으로 만들어진 함수
- 씬이 실행 된 직후 1회 호출됨

■Start()

- 데이터를 초기화하는 목적으로 만들어진 함수
- 첫 프레임 업데이트가 실행되기 직전에 1회 호출됨

■OnEnable()

- 컴포넌트가 비활성되었다가 다시 활성화될 때 마다 1회 호출됨

- Start()와 OnEnable()은 컴포넌트가 비활성화 되어있으면 함수를 호출하지 않음
- Awake()는 컴포넌트가 비활성화 되었어도 호출함

이벤트 함수

```
Unity 메시지 | 참조 0개
void Update()
{
    Debug.Log("Update() 호출");
}
참조 0개
void LateUpdate()
{
    Debug.Log("LateUpdate() 호출");
}

private void FixedUpdate()
{
    Debug.Log("FiexedUpdate() 호출");
}
```

업데이트 함수
스크립트와 결과

■Update()

- 현재 씬이 실행된 후 매 프레임마다 호출됨

■LateUpdate()

- 현재 프레임에서 모든 오브젝트의 Update()가 호출된 후 호출됨

■FixedUpdate()

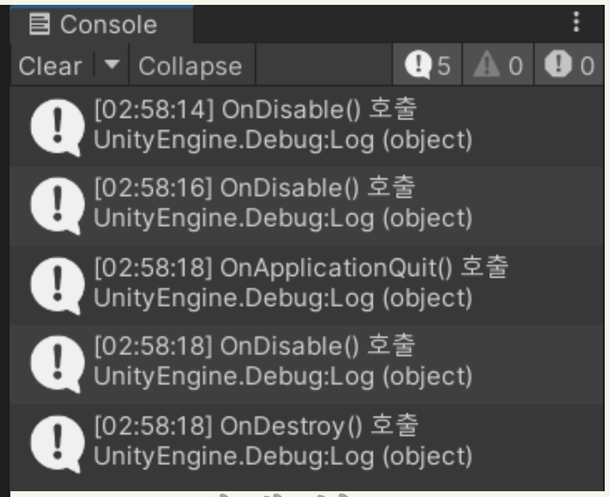
- 정해진 횟수만큼 호출됨
- 기본값은 0.02로 1초이 50회씩 호출됨
  Edit->Project Settings->Time->Tomestep 에서 호출 주기 설정 가능

- 세 함수 모두 게임오브젝트와 컴포넌트가 활성화 되어있을 때 호출됨

이벤트 함수

```
Unity 메시지 | 참조 0개
void OnDestroy()
{
    Debug.Log("OnDestroy() 호출");
}
참조 0개
void OnApplicationQuit()
{
    Debug.Log("OnApplicationQuit() 호출");
}

private void OnDisable()
{
    Debug.Log("OnDisable() 호출");
}
```



해체 함수

스크립트와 결과

■OnDestroy()

- 게임오브젝트가 파괴될 때 1회 호출됨
- 씬 변경이나 게임이 종료될 때 오브젝트가 파괴되기 때문에 호출됨

■OnApplicationQuit()

- 게임이 종료될 때 1회 호출됨
- 에디터에서는 플레이 모드 중지로 테스트 가능함

■OnDisable()

- 컴포넌트가 비활성화 될 때 마다 1회 호출됨

- OnDestroy()와 OnApplicationQuit()는 컴포넌트가 비활성화 되어있어도 함수가 호출됨
- OnDisable()은 컴포넌트가 활성화 되어있어야 함

Thanks