2D Interactive Contents
(Chapter 4)

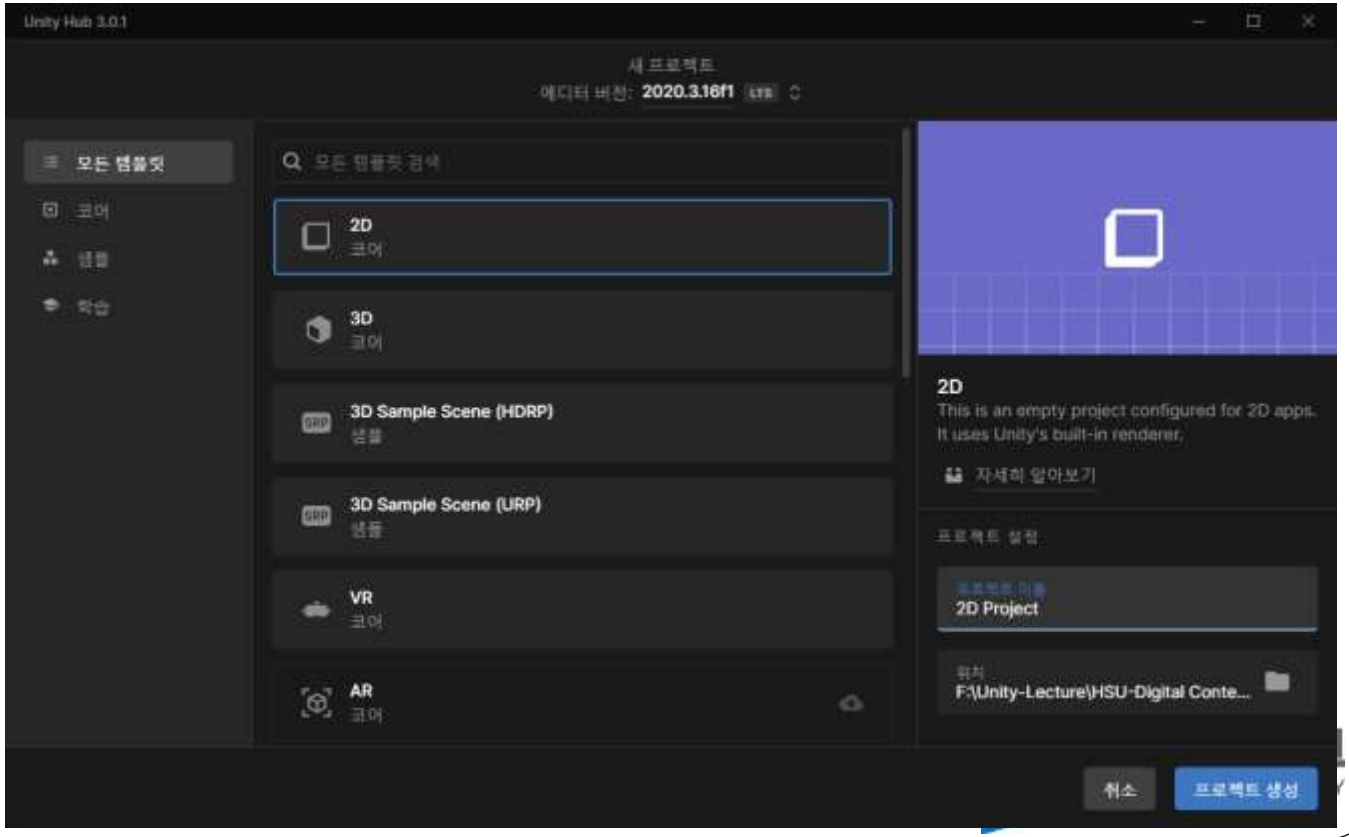
Jin-Mo Kim
jinmo.kim@hansung.ac.kr

2D 콘텐츠 제작

- 2D 콘텐츠 제작 과정
  - 미니 2D 게임 콘텐츠를 제작하는 과정을 통해 2D 콘텐츠에서 사용되는 용어나 기법을 학습
    - 스프라이트를 가져오고 나누는 방법
    - 2D 콘텐츠 제어를 위한 스크립트 구현 방법
    - 2D 콘텐츠에서의 프리팹(Prefab) 활용 방법

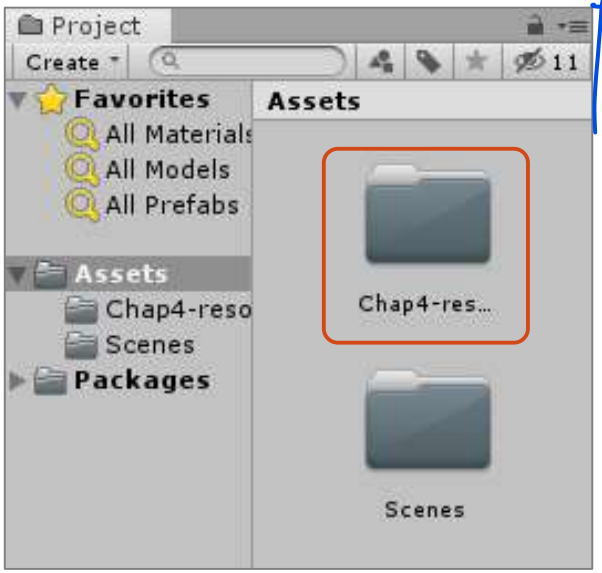
2D 콘텐츠 제작

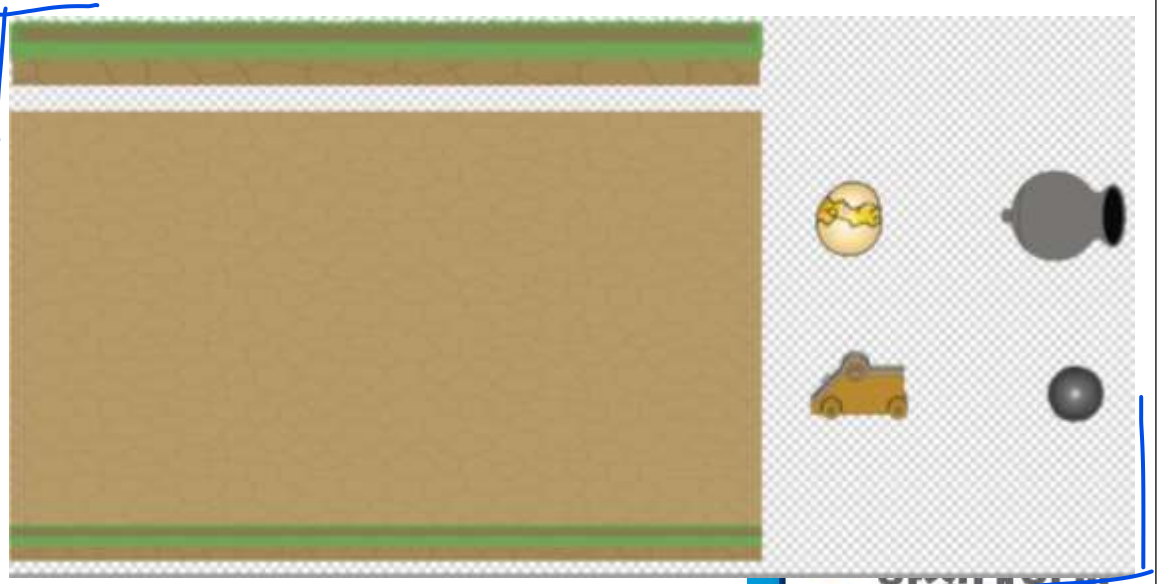
- 프로젝트 생성
  - 2D 프로젝트 생성
    - 2D 프로젝트는 신(Scene) 뷰가 2D에 맞게 정면을 바라보도록 자동 설정
    - X-Y 2D 평면에서 작업을 설정



2D 콘텐츠 제작

- 리소스 추가
  - 2D 게임 콘텐츠 사용될 이미지 리소스를 프로젝트에 추가
    - Chap4-resources 폴더 전체를 프로젝트로 Drag & Drop
    - 스프라이트
      - 2D 콘텐츠의 배경, 캐릭터 등의 목적으로 사용되는 이미지 파일
      - psd, tiff, jpg, tga, png, gif, bmp, iff 등의 이미지 포맷을 지원 by 유니티
      - 단, 이미지 데이터는 여백을 투명하게 만든 상태에서 사용해야 함

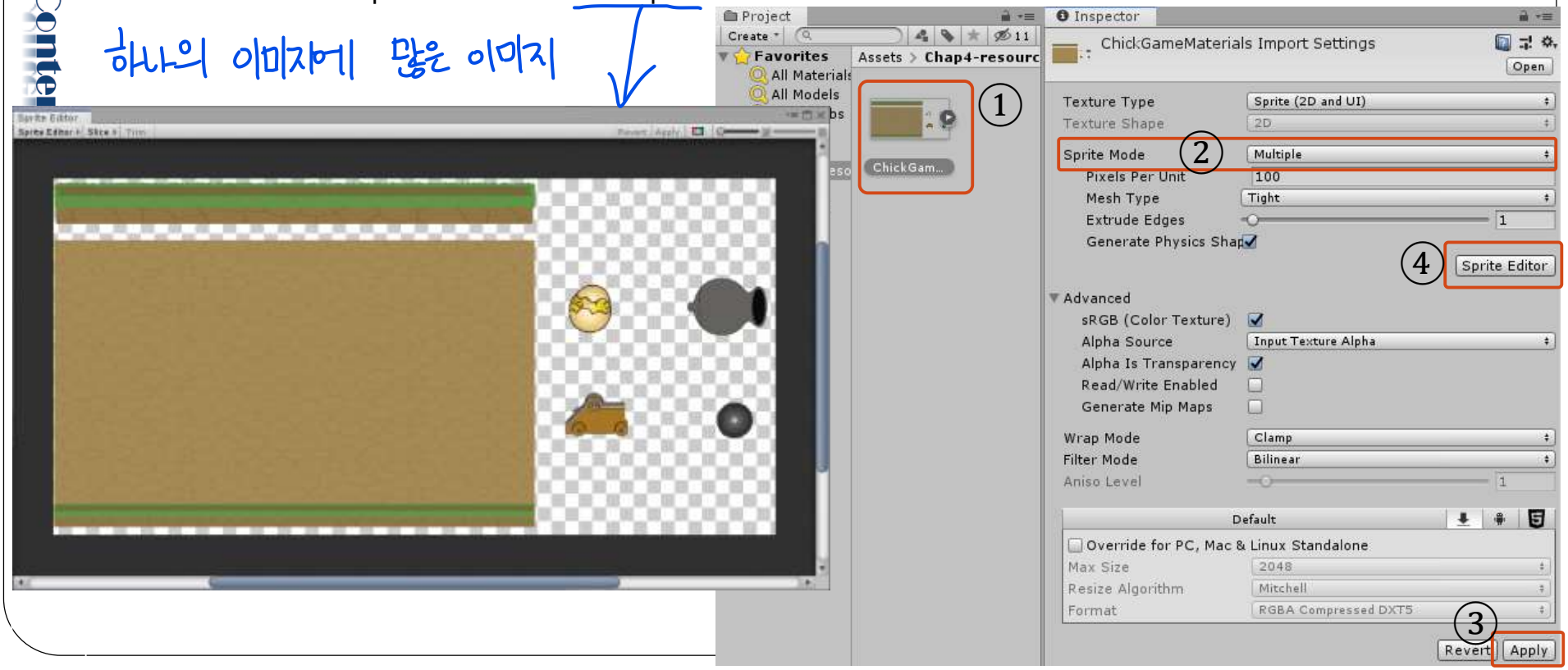




2D 콘텐츠 제작

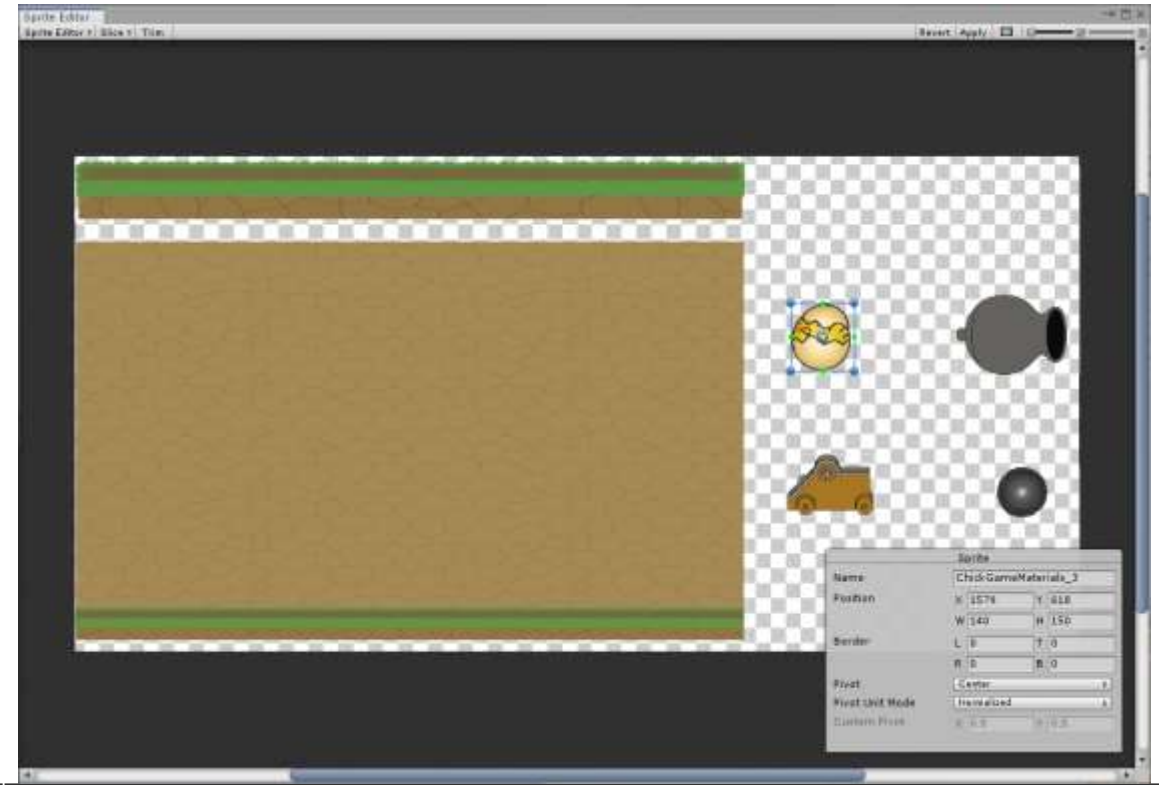
- 스프라이트 편집
  - 스프라이트 에디터(Sprite Editor)를 활용한 스프라이트 이미지 편집
    - 하나의 데이터 안에 스테이지, 배경, 대포 등이 하나로 묶여있음
    - 각각의 이미지로 나눠주는 작업이 필요 → 스프라이트 에디터
      - Sprite Mode → Multiple로 변경

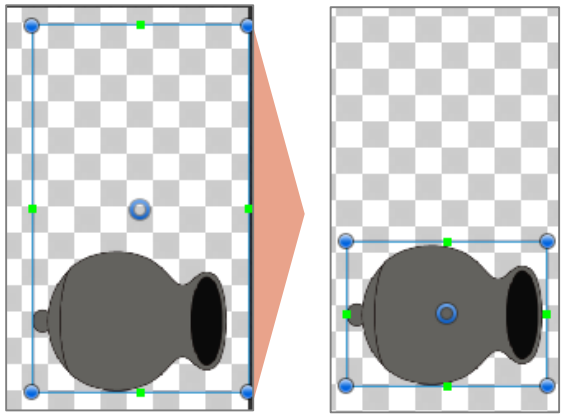
하나의 이미지에 많은 이미지



2D 콘텐츠 제작

- 스프라이트 편집
  - 스프라이트 나누기
    - 스프라이트 에디터 → Slice → Type → Automatic
      - Slice 버튼 클릭: 자동으로 잘려지는 것을 확인→ 스프라이트 이름 자동 설정
      - 자동으로 자를 경우 이미지에 딱 맞게 잘리지 않는 경우도 발생 → 직접 수정



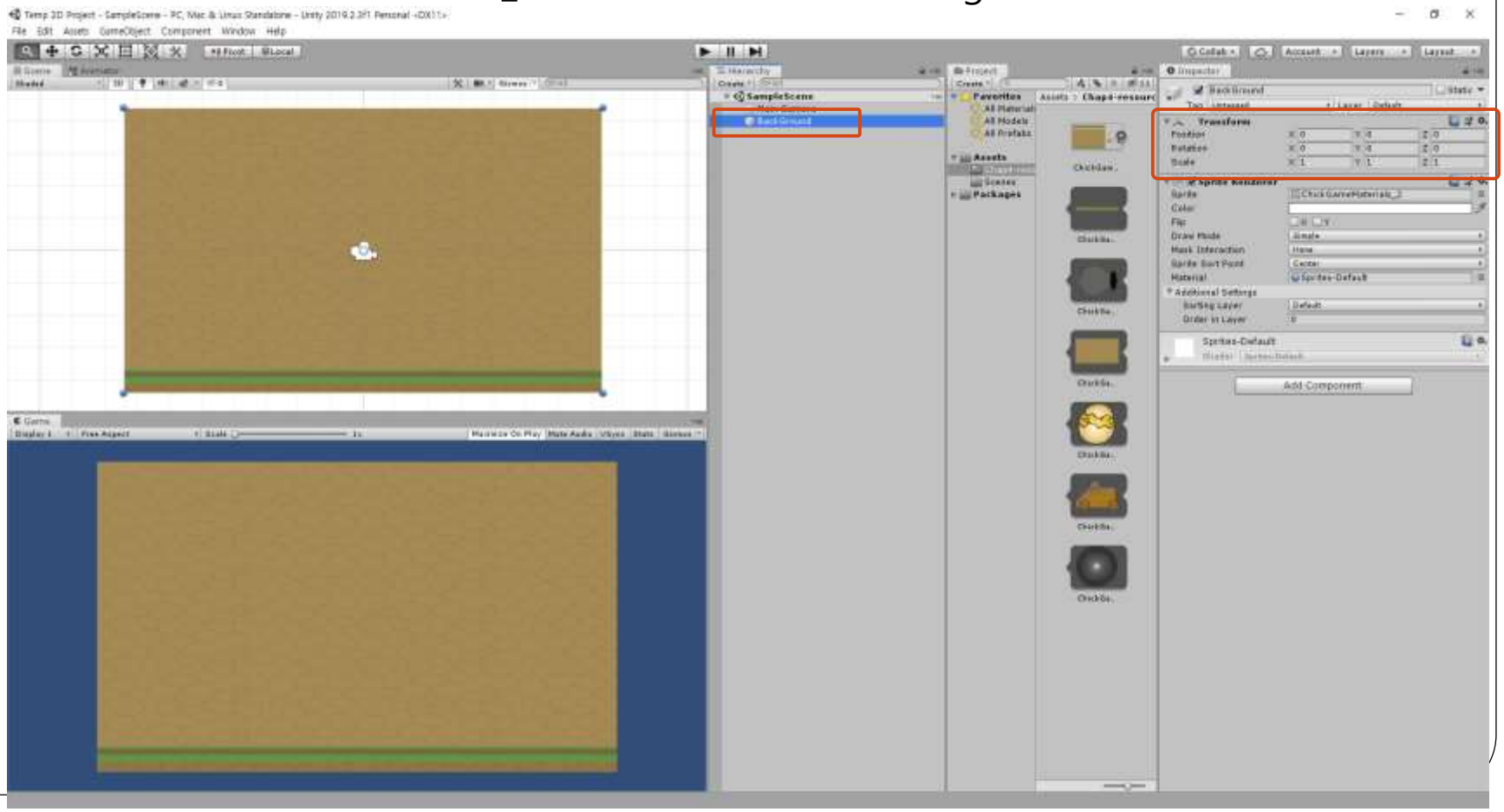


2D 콘텐츠 제작

- 스프라이트 편집
  - 스프라이트 나누기
    - ChickGameMaterial_0~5 까지 스프라이트 나눠서 설정됨

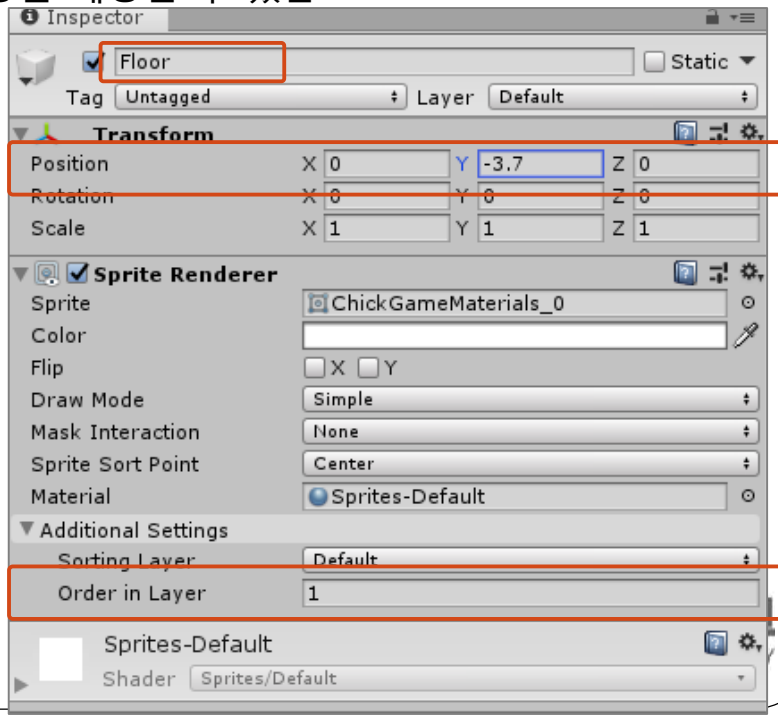
2D 콘텐츠 제작

- 스테이지 생성
  - 배경 배치
    - ChickGameMaterial_2를 배경으로 사용 → Background로 이름 변경



2D 콘텐츠 제작

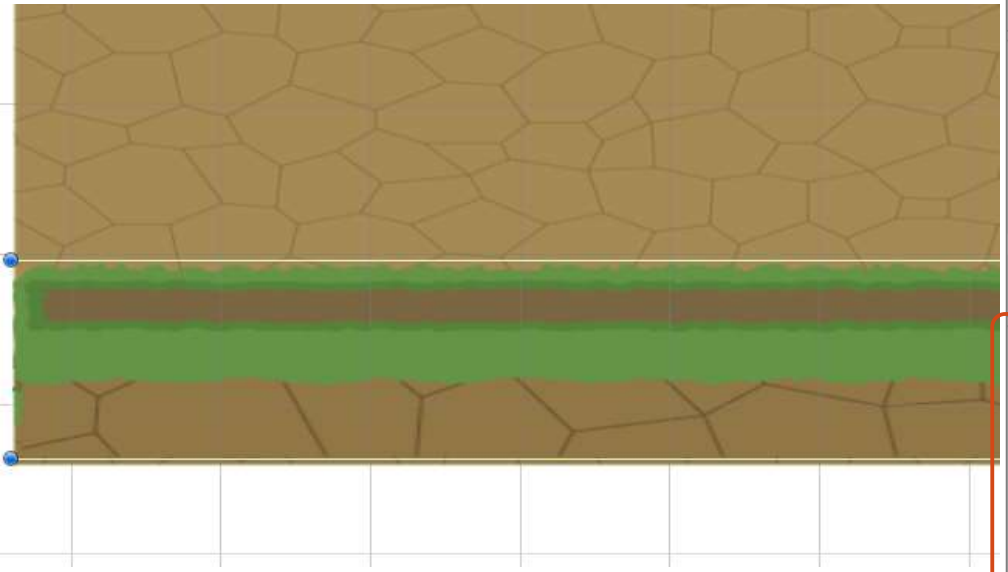
- 스테이지 생성
  - 바닥 배치
    - ChickGameMaterial_0를 바닥으로 사용 → Floor로 이름 변경
      - Position : 0, -3.7, 0 으로 이동
      - Order in Layer : 1로 수정
      - Order in Layer → 이미지를 겹쳐서 여러 장 그리는 경우 그려질 순서를 지정해야 배경 뒤로 다른 이미지가 가려지는 현상을 예방할 수 있음
        - 값이 클 수록 앞쪽에 그려짐!

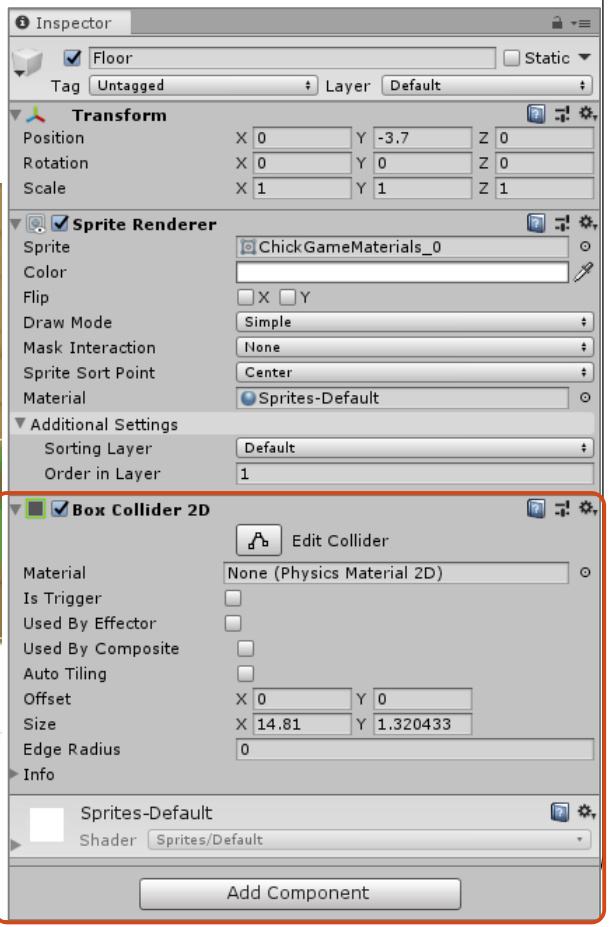


: 클릭의 레이어, 꼭 신경쓰자 →

2D 콘텐츠 제작

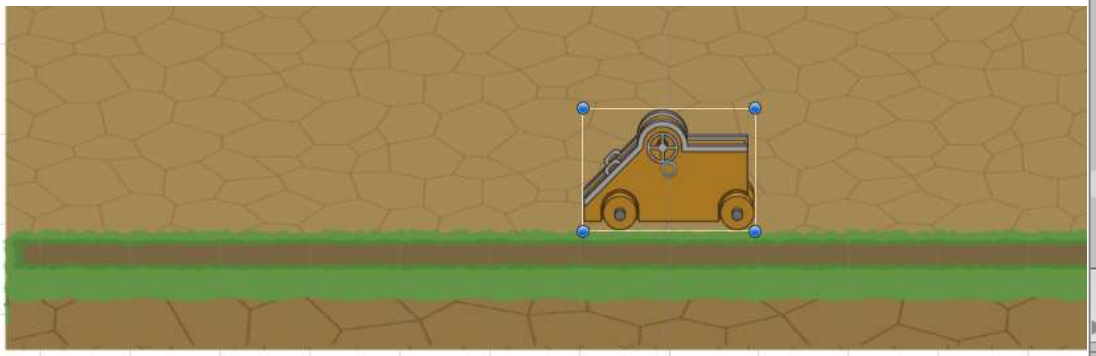
- 스테이지 생성
  - 바닥 충돌 설정
    - Floor 오브젝트에 충돌체 설정
      - Add Component → Physics2D → Box Collider 2D
      - 녹색 사각형 충돌체가 생성

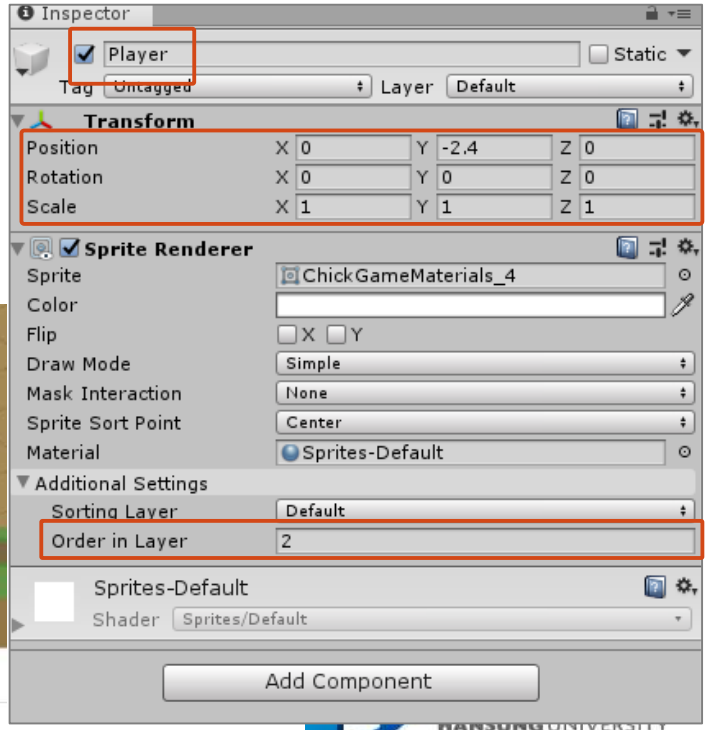




2D 콘텐츠 제작

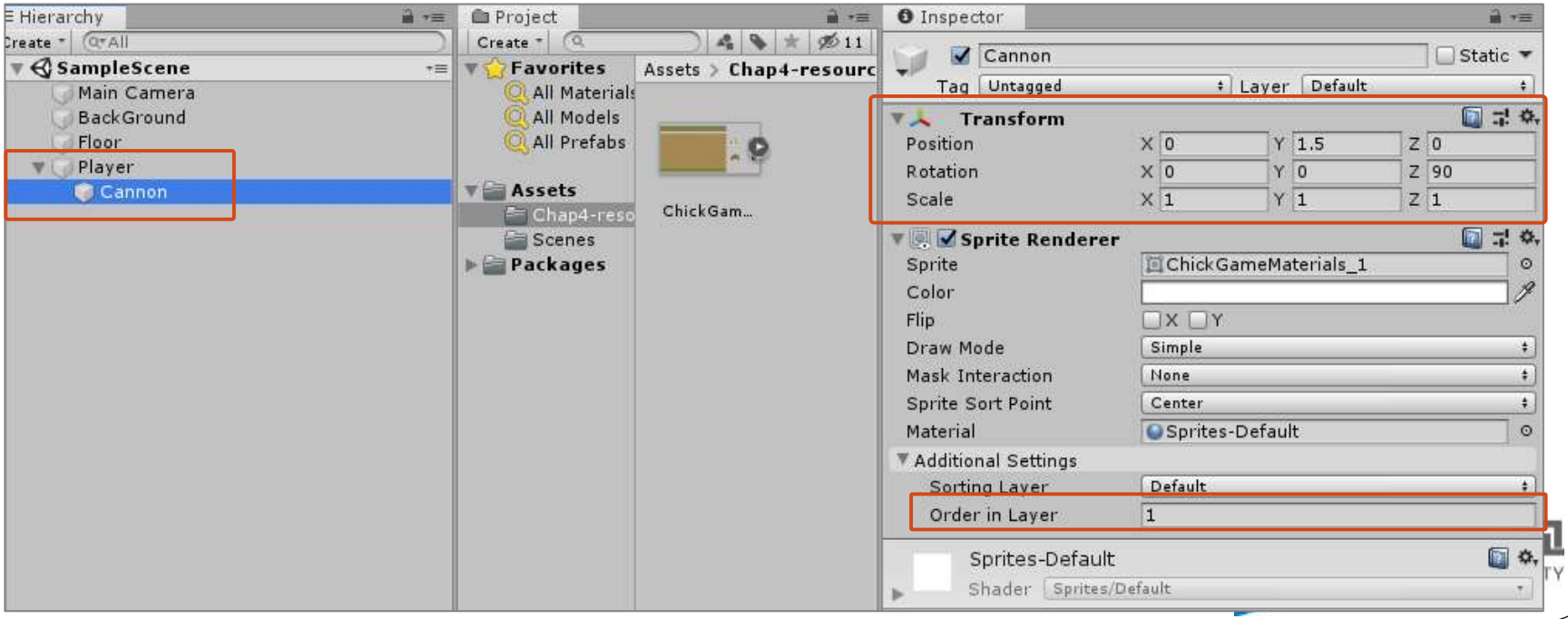
- 플레이어 설정
  - 플레이어 배치
    - ChickGameMaterial_4 배치 → Player로 이름 변경
      - Position: 0, -2.4, 0 이동
      - Order in Layer: 2로 변경





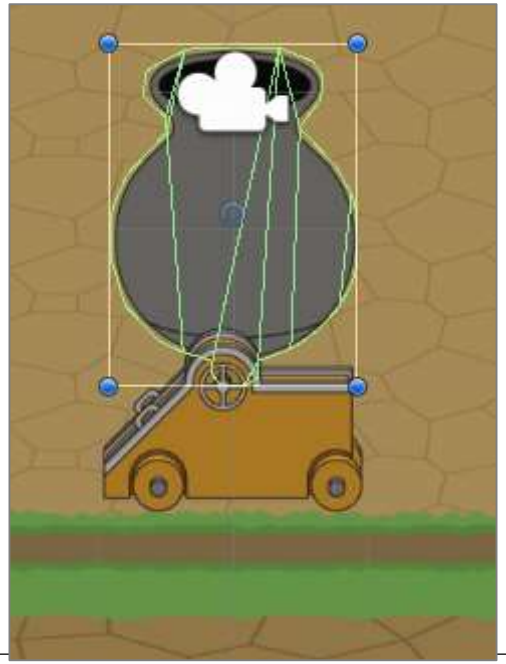
2D 콘텐츠 제작

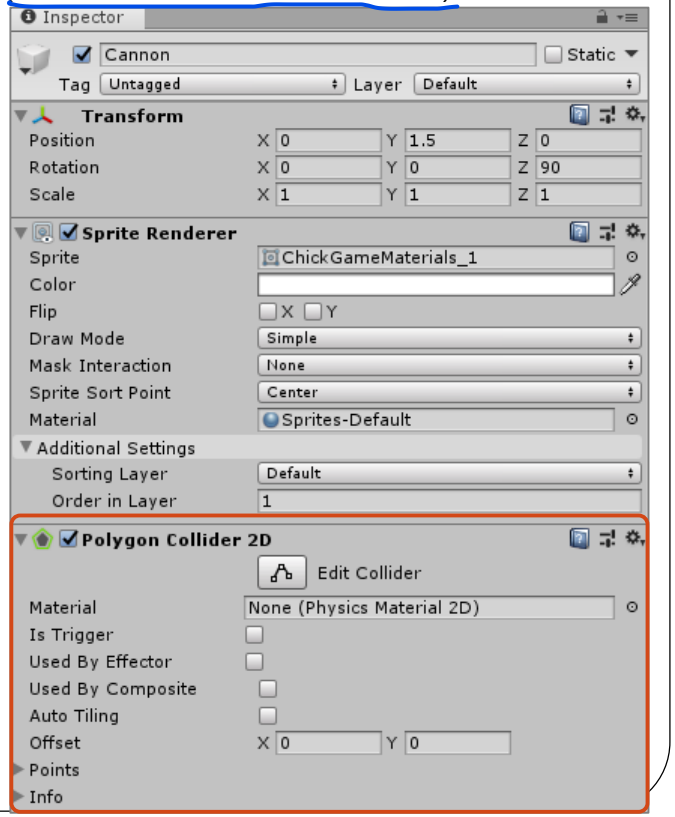
- 플레이어 설정
  - 플레이어 추가요소 배치
    - ChickGameMaterial_1 배치 → Cannon으로 이름 변경
      - Order in Layer: 1로 변경
    - Cannon을 Player의 자식 객체로 설정 후 Player를 기준으로 이동, 회전
      - Position: 0, 1.5, 0
      - Rotation: 0, 0, 90



2D 콘텐츠 제작

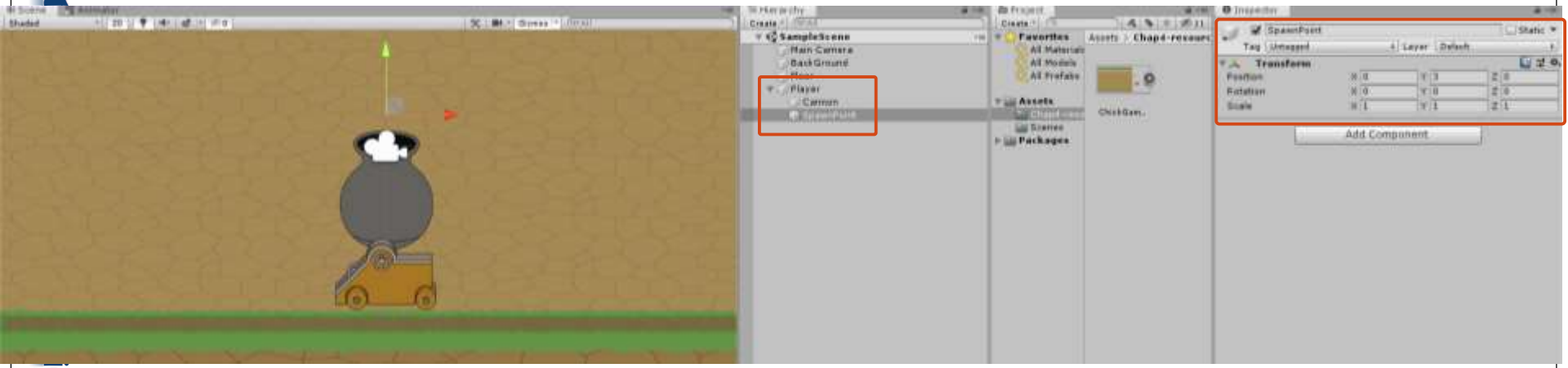
- 플레이어 설정
  - 캐논에 충돌체 설정
    - 충돌 판정을 위한 Collider 추가
      - Add Component → Physics 2D → Polygon Collider 2D
        - 그림과 일치하는 폴리곤을 계산하여 충돌체를 생성 (가장 정확한 충돌 검사가 가능)





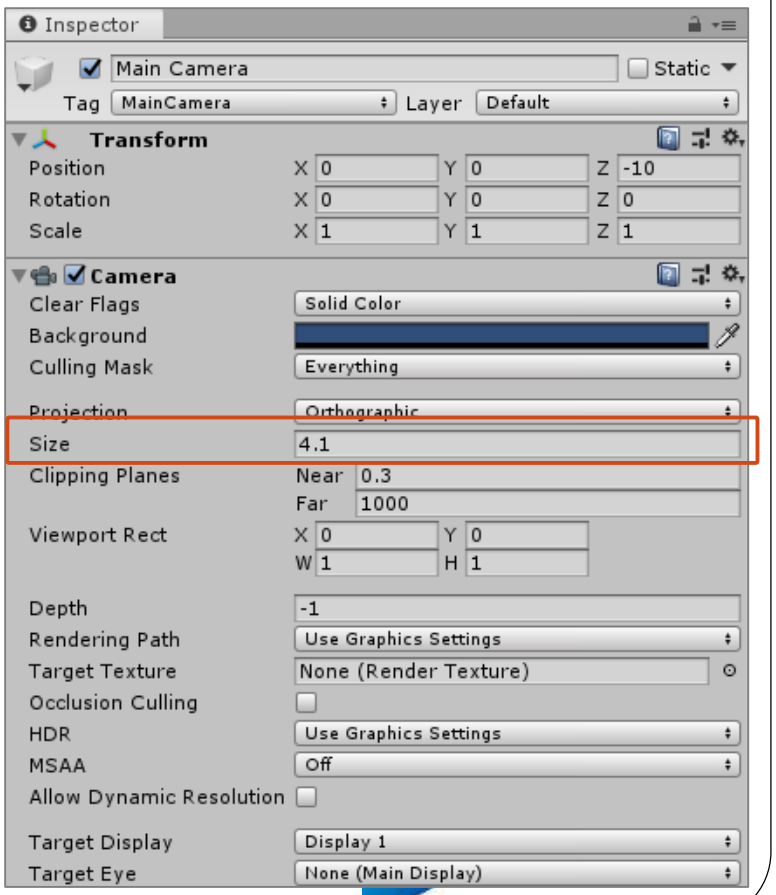
2D 콘텐츠 제작

- 플레이어 설정
  - 발사대 배치
    - 포탄을 발사하기 위한 발사대를 배치
      - 발사대는 눈에 보이는 객체가 아닌 포탄이 발사대는 위치를 지정
      - Hierarchy → Create Empty
        - 이름: SpawnPoint
        - Player의 자식 객체로 설정 후
        - Position: 0, 3, 0 (Player를 기준으로 위치 설정)



2D 콘텐츠 제작

- 카메라 설정
  - 카메라의 범위를 배경과 일치하도록 조정
    - Main Camera → Size: 4.1
    - Game 뷰 → 비율: 16:9



2D 콘텐츠 제작

- 플레이어 제어
  - 스크립트 생성
    - Scripts 폴더 생성
      - Project → Create → C# Script
        - 이름: cshPlayerController.cs

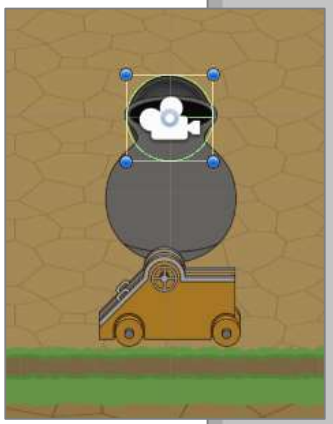
2D 콘텐츠 제작

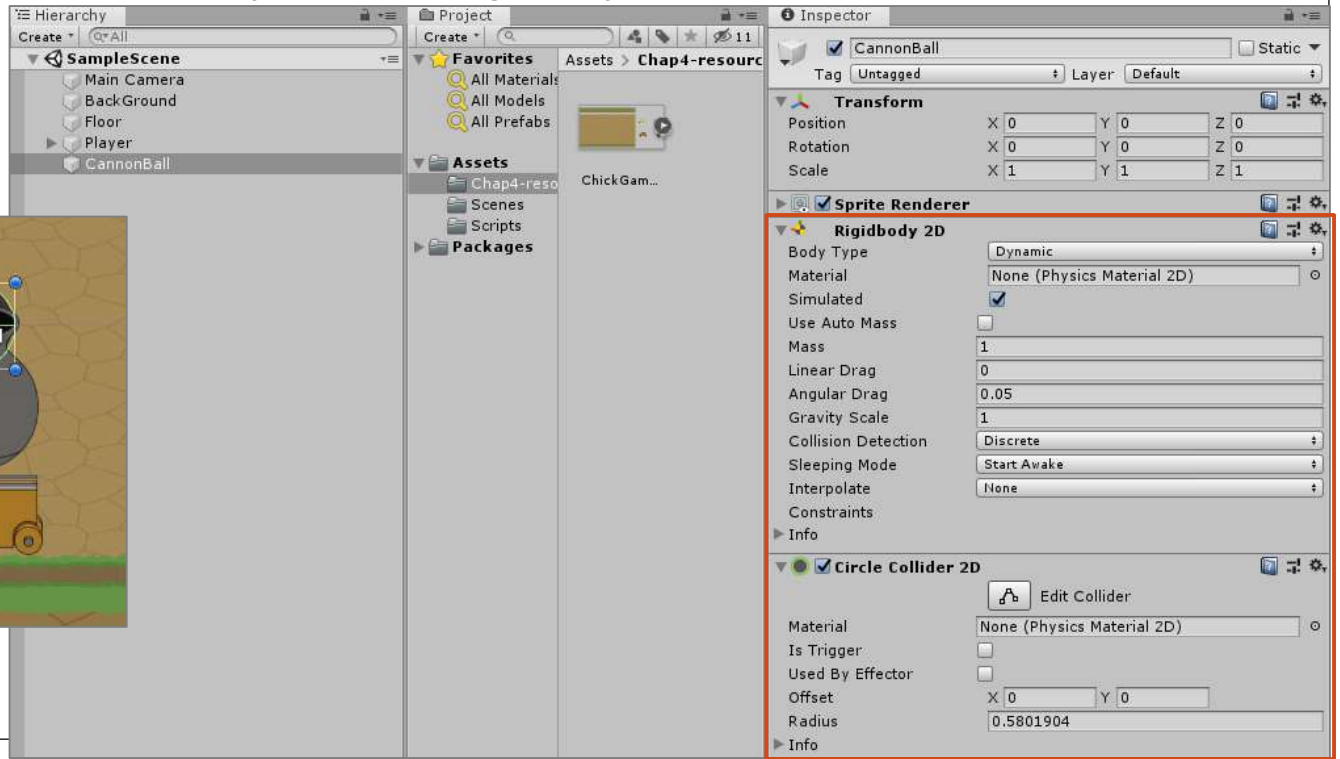
- 플레이어 제어
  - 스크립트를 Player 객체에 등록

```
public class cshPlayerController : MonoBehaviour
{
    public float speed = 8f;//Player의 이동 속도
    public float moveableRange = 5.5f; // 이동 가능한 범위
    public float power = 1000f; // CannonBall을 발사하는 힘

    void Update()
    {
        //Player 이동 (이동 범위를 movableRange로 제한)
        transform.Translate(Input.GetAxisRaw("Horizontal") * speed * Time.deltaTime, 0, 0);
        transform.position
        = new Vector2(Mathf.Clamp(transform.position.x, -moveableRange, moveableRange), transform.position.y);
    }
}
```

2D 콘텐츠 제작

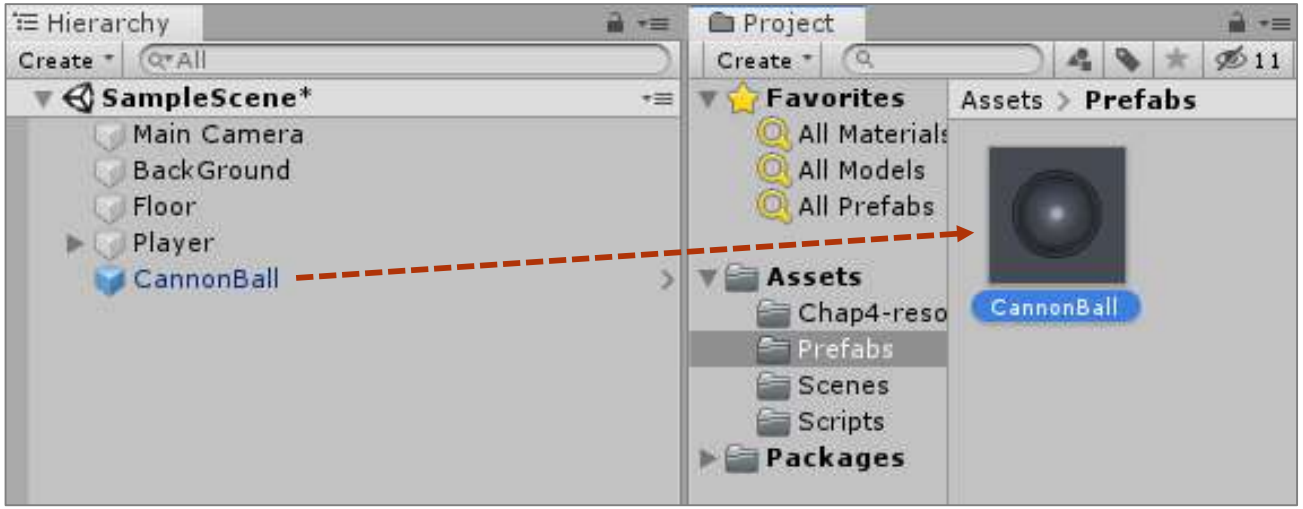
- 포탄 발사
  - 프리팹을 사용하여 포탄 발사 기능 구현
    - ChickGameMaterial_5 배치 → CannonBall으로 이름 변경
      - Order in Layer: 1로 변경
    - 물리 적용과 충돌 설정
      - Add Component → Physics 2D → Rigidbody 2D, Circle Collider 2D





2D 콘텐츠 제작

- 포탄 발사
  - 프리팹을 사용하여 포탄 발사 기능 구현
    - 프리팹 등록
      - Project → Create → Folder 이름: Prefabs
        - 드래그 앤 드롭으로 CannonBall을 Prefabs 폴더 안으로 이동
      - Hierarchy에 있는 CannonBall을 삭제



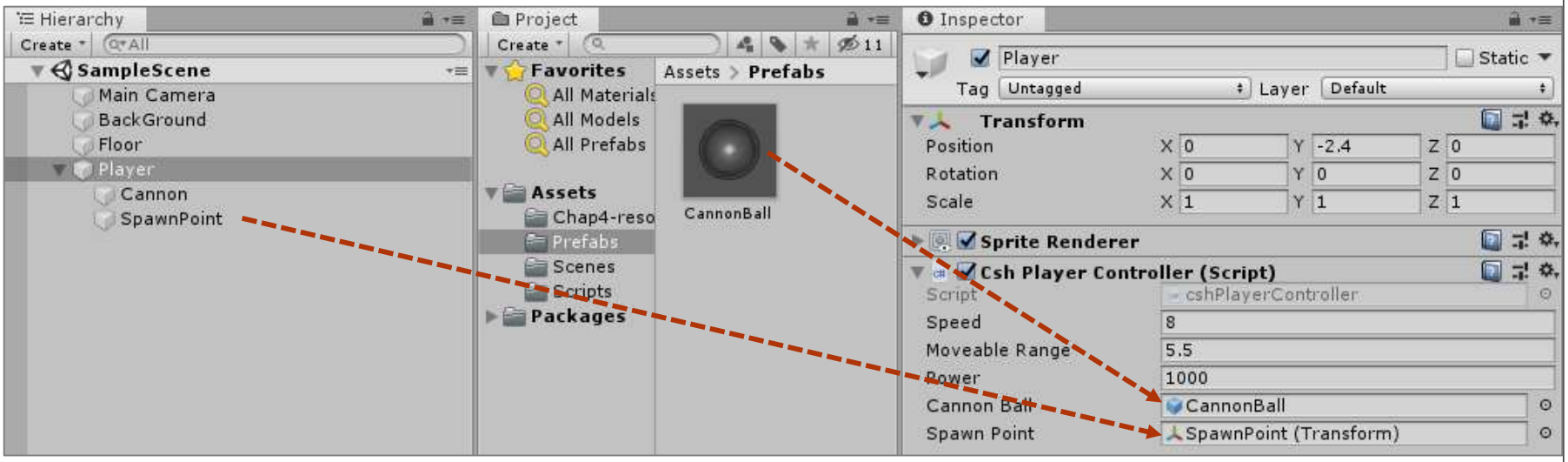
2D 콘텐츠 제작

- 포탄 발사
  - 포탄 발사 스크립트 작성
    - cshPlayerController 스크립트 수정

```
public class cshPlayerController : MonoBehaviour
{
    // 포탄 발사 변수 추가
    public GameObject cannonBall; //Player에서 발사할 CannonBall
    public Transform spawnPoint; //Cannon 발사 지점

    void Update()
    {
        // 포탄 발사 기능 추가
        if (Input.GetKeyDown(KeyCode.Space))  //스페이스키 버튼 이벤트 처리
        {
            Shoot(); //Shoot함수 호출
        }
    }

    void Shoot()
    {
        //새 cannonBall을 생성하여 newBullet에 할당
        GameObject newBullet = Instantiate(cannonBall, spawnPoint.position, Quaternion.identity) as GameObject;
        //newBullet의 Rigidbody2D를 참조하여 AddForce 함수로 물리적으로 발사
        newBullet.GetComponent<Rigidbody2D>().AddForce(Vector3.up * power);
    }
}
```

2D 콘텐츠 제작

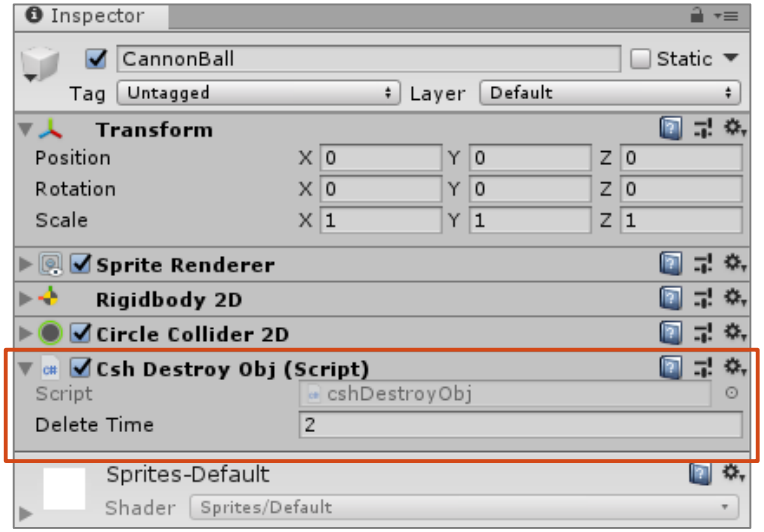
- 포탄 발사
  - 포탄 발사 스크립트 작성
    - 스크립트의 전역 변수에 대응되는 객체 설정



2D 콘텐츠 제작

- 포탄 발사
  - 포탄 스크립트추가
    - 일정시간이 경과하면 포탄이 자동으로 사라지는 기능 구현
      - Scripts 폴더 안에 새로운 스크립트 추가
      - Project → Create → C# Script : cshDestroyObj.cs
      - 스크립트를 CannonBall 프리팹 오브젝트에 등록

```
public class cshDestroyObj : MonoBehaviour
{
    //오브젝트를 제거하는 인터벌
    public float deleteTime = 2.0f;

    void Start()
    {
        //오브젝트를 생성한 후 deleteTime 만큼 시간이 경과하면 제거
        Destroy(gameObject, deleteTime);
    }
}
```

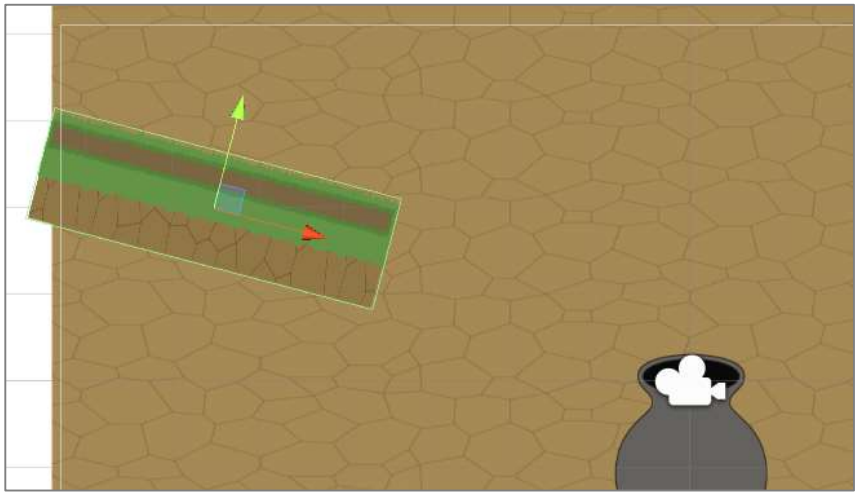


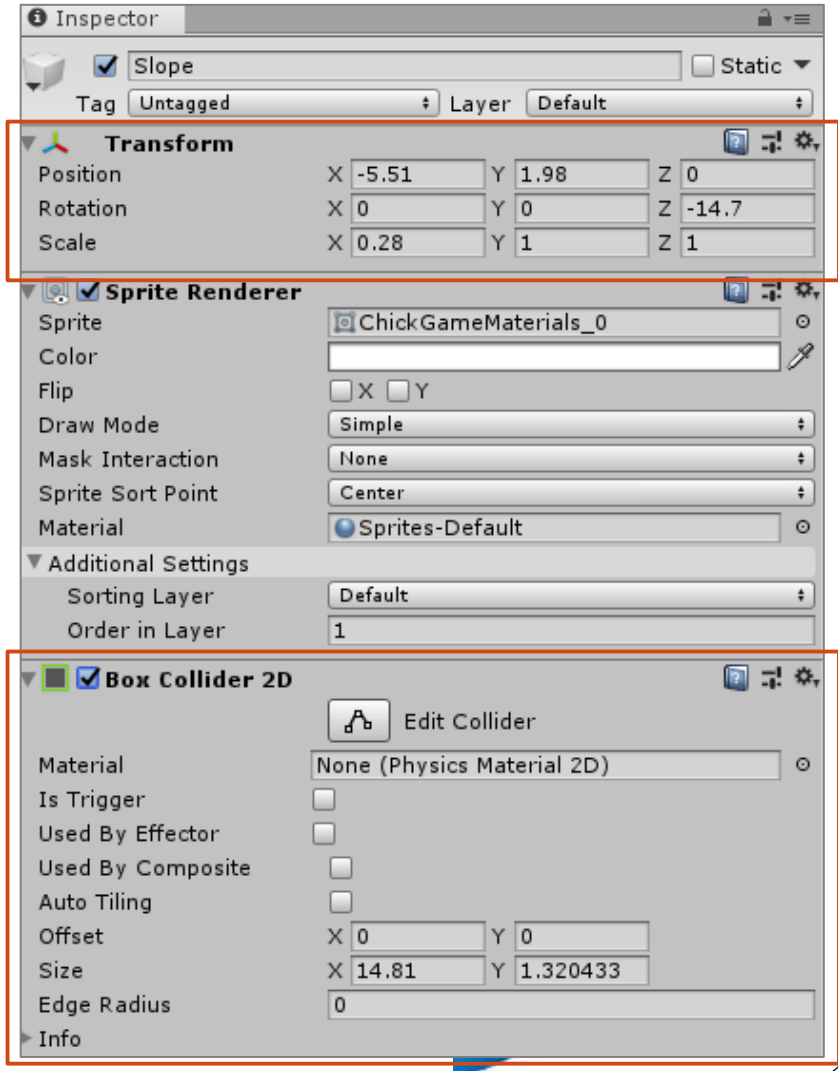
2D 콘텐츠 제작

- 병아리 구슬 제작
  - 일정 시간 주기로 생성되어 굴러 떨어지는 병아리 구슬
  - 경사면 배치
    - ChickGameMaterial_0 배치 → Slope으로 이름 변경
      - Order in Layer: 1로 변경
      - Position: -5.51, 1.98, 0
      - Rotation: 0, 0, -14.7
      - Scale: 0.28, 1, 1
    - 병아리 구슬이 지면을 타고 내려올 수 있도록 충돌체 속성 추가
      - Slope 객체 선택 → Add Component → Box Collider 2D

2D 콘텐츠 제작

- 병아리 구슬 제작



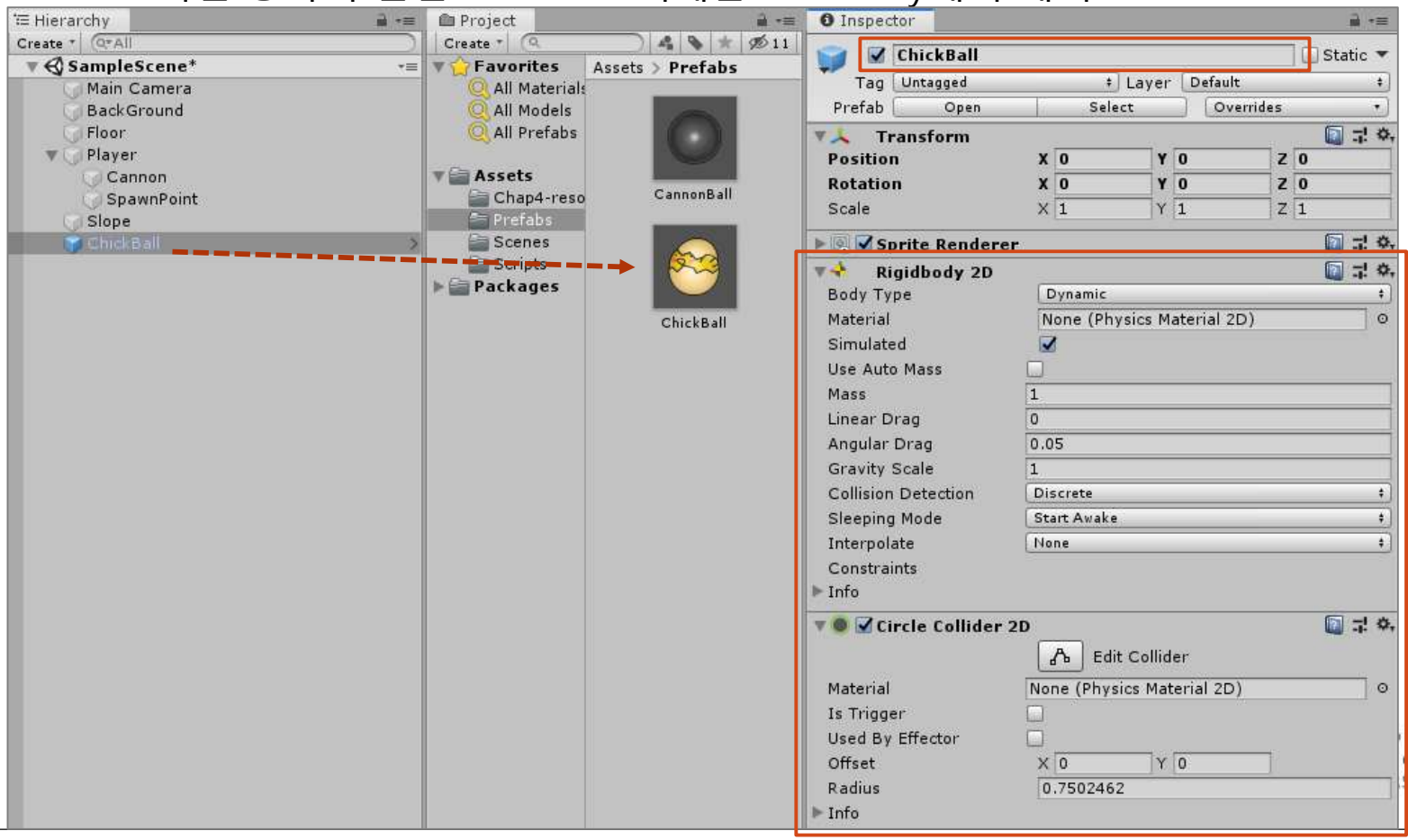


2D 콘텐츠 제작

- 병아리 구슬 제작
  - 병아리 구슬
    - ChickGameMaterial_3 배치 → ChickBall으로 이름 변경
      - Order in Layer: 2로 변경
    - 물리 적용과 충돌 설정
      - Add Component → Physics 2D → Rigidbody 2D, Circle Collider 2D
    - 프리팹 등록
      - Prefabs 폴더에 드래그 앤 드롭으로 프리팹 등록

2D 콘텐츠 제작

- 병아리 구슬 제작
  - 병아리 구슬
    - 프리팹 등록이 끝난 ChickBall 객체는 Hierarchy에서 제거



2D 콘텐츠 제작

- 병아리 구슬 제작
  - 병아리 구슬 스크립트
    - 일정시간 간격으로 병아리 구슬이 생성되는 스크립트 추가
      - Scripts 폴더 안에 Project → Create → cshChickGenerator.cs 생성

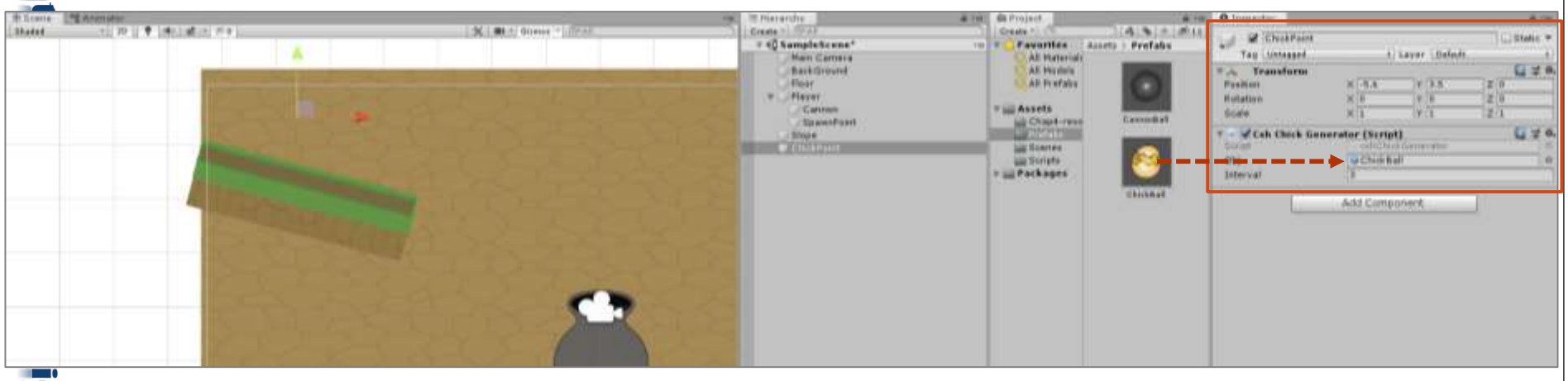
```
public class cshChickGenerator : MonoBehaviour
{
    public GameObject obj; //ChickBallPrefab 설정
    public float interval = 3.0f; //다음에 함수가 호출될 인터벌

    void Start()
    {
        //SpawnObj함수를 게임이 실행된 0.1초 후에 호출, 이후 interval초 마다 호출 된다.
        InvokeRepeating("SpawnObj", 0.1f, interval);
    }

    //SpawnObj함수는 ChickBallPrefab을 생성한다.
    void SpawnObj()
    {
        Instantiate(obj, transform.position, transform.rotation);
    }
}
```

= 총 3.1 초후

프리팹 복제 (병아리), 복제 생성위치

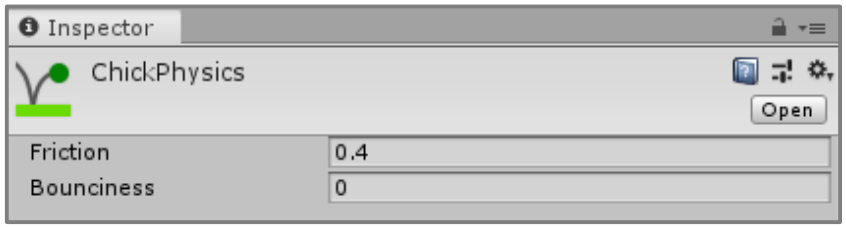
2D 콘텐츠 제작

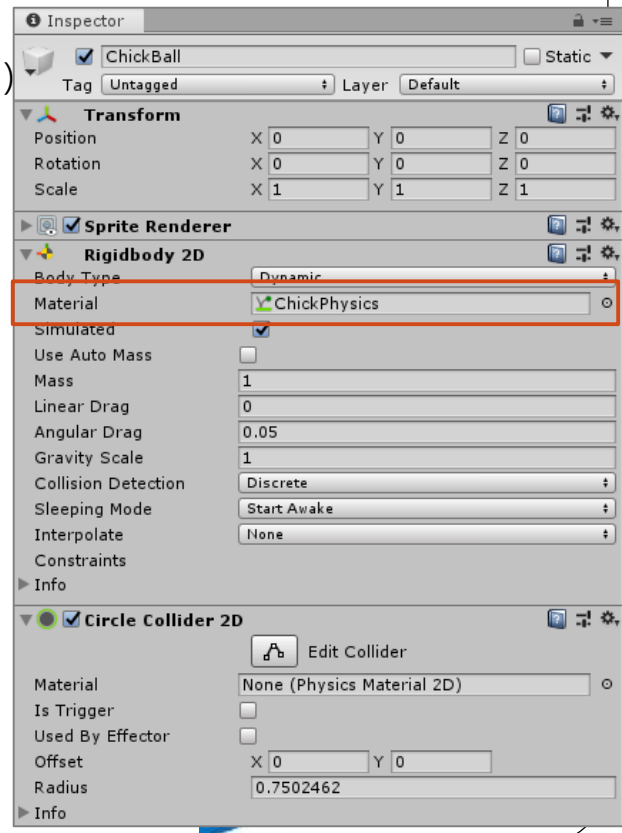
- 병아리 구슬 제작
  - 병아리 구슬 생성 위치 설정
    - Hierarchy → Create → Create Empty
      - 이름: ChickPoint
      - Position: -5.6, 3.5, 0
    - cshChickGenerator 스크립트를 추가
      - Obj 객체에 ChickBall 프리팹을 등록



2D 콘텐츠 제작

- 기능 개선
  - 병아리 구슬에 탄성 추가
    - Project → Create → Physics Material 2D
      - 이름 : ChickPhysics
      - Friction: 0.4 (마찰계수, 0: 마찰이 없는 상태)
      - Bounciness : 1 (반발계수, 0: 탄성이 전혀 없는 상태)
    - 생성한 물리 속성을 ChickBall 프리팹에 등록
      - Rigidbody 2d → Material 추가





2D 콘텐츠 제작

- 기능 개선
  - 병아리 구슬 일정 시간 뒤 제거
    - cshDestroyObj 스크립트를 ChickBall 프리팹에도 등록
      - Delete Time을 10으로 수정