

〈Corgi Engine〉

1. 한 줄 요약 / 핵심 장점

Corgi Engine은 Unity 전용 완성형 2D & 2.5D 플랫폼러 킷으로, 빠르고 타이트한 논-피직스 기반 캐릭터 컨트롤러(이동·점프·대시·월점프 등), 방대한 능력(Abilities) 시스템, 레벨/데모/툴을 포함해 실전 게임 제작에 바로 쓸 수 있는 패키지입니다. ([링크](#))[1])



2. 적용 대상

- 플랫폼 액션/메트로베니아/윙스크롤 게임을 빠르게 완성하려는 팀
- 모바일(터치)·로컬 멀티플레이 대응이 필요한 프로젝트
- 캐릭터 움직임(물리 아님)이 정밀해야 하는 게임(“tight controls”) >

빠른 프로토타입부터 상업 릴리즈까지 모두 적합. ([링크](#))[1])



3. 설치 & 시작 (간단 가이드)

3.1 구매 / 다운로드: Unity Asset Store에서 Corgi Engine 구매 후 Import. (패키지 매니저의 My Assets에서도 가져오기 권장). ([\[Unity Asset Store\]](#)[2])

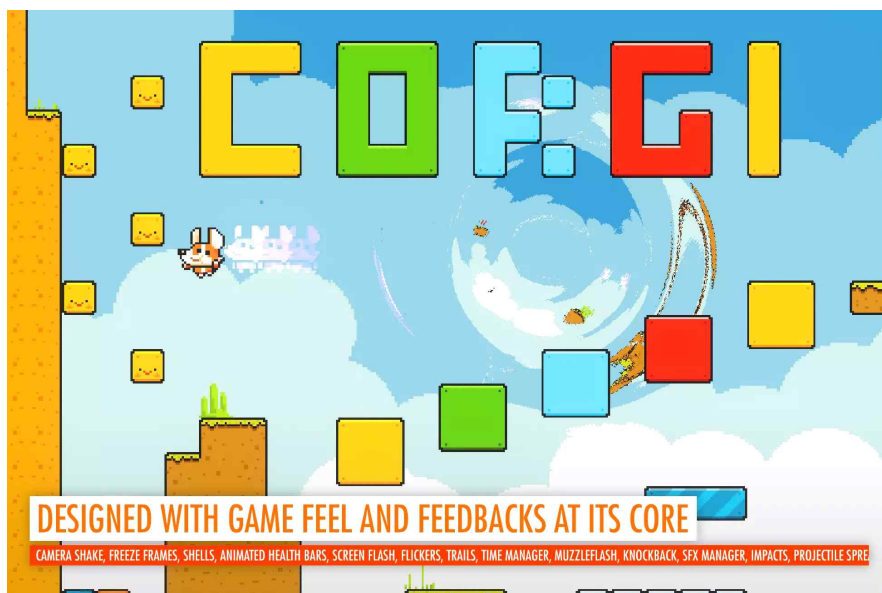
3.2 빈 프로젝트 권장: 설치 시 프로젝트 설정이 덮어써질 수 있으므로 새 빈 프로젝트에 임포트 권장(문서 권고). ([\[링크\]](#)[3])

3.3 데모 실행: 패키지에 포함된 데모 씬(예: MinimalLevel, 여러 데모 레벨)을 열어 동작을 확인. ([\[링크\]](#)[1])



4. 핵심 구성요소(개념 정리)

- Corgi Controller / Character: 엔진의 핵심 캐릭터 클래스(충돌·중력·상태머신 관리). 각 캐릭터는 여러 Character Ability(이동, 점프, 대시 등)를 조합해 동작함. ([링크](#))[4])
- Character Abilities: Jump, Dash, WallJump, Jetpack, Crouch, Run, Push, Damage 등 수십 종류의 재사용 가능한 컴포넌트(스크립트). 각 Ability는 독립적으로 켜고 끌 수 있고, 인스펙터로 세부 파라미터 조정 가능. ([링크](#))[5])
- Level Manager: 씬/레벨의 규칙(스폰, 체크포인트, 엔트리 포인트)을 담당. 플레이어 스폰, 라이프/리스폰 정책 등 중앙제어. ([링크](#))[6])
- 데모 레벨 & 에셋: 60개 이상의 데모 레벨, GUI, 샘플 에셋 포함(레벨 연결/메뉴/로딩 등 예제 포함). ([링크](#))[7])



5. 주요 기능(요약)

- 정밀한 2D/2.5D 논-피직스 캐릭터 컨트롤러(빠른 반응성)
- 다양한 기본 Ability (Jump, DoubleJump, WallJump, Dash, Jetpack, Push, Swim 등)
- 애니메이션 파라미터 통합 (Mecanim 연동)
- MMFeedbacks(피드백 시스템)로 이펙트/카메라 셰이크 등 손쉬운 게임피드 추가
- Tilemap·Physics2D·Cinemachine 등 Unity 생태계와 호환성 높음
- 모바일 입력, 로컬 멀티플레이(로컬 2P 등) 지원
(자세한 능력·클래스 리스트는 공식 API 문서 참조). ([링크](#))[8])

HIGH QUALITY, EASY TO USE, GREAT STARTING POINT TO LEARN

REAL INDUSTRY EXPERIENCE, MODULAR AND READABLE CODE, FULL DOCUMENTATION, VIDEO TUTORIALS, CONSTANT UPDATES, GREAT SUPPORT (ACCORDING TO THE REVIEWS)

6. 실무 설정 / 워크플로우 (권장 단계)

6.1 프로토타입 캐릭터 만들기

- ‘Characters/’에 있는 샘플 플레이어 프리팹 복사) 씬에 배치.

- 인스펙터에서 원하는 Ability(예: CharacterJump, CharacterDash, CharacterWalljump 등)를 확인/ON-OFF. ([링크](#))[4])

6.2 입력 매핑

- Corgi의 InputSettings에서 키/버튼(또는 모바일 터치) 매핑 확인 및 수정.

6.3 레벨 구성

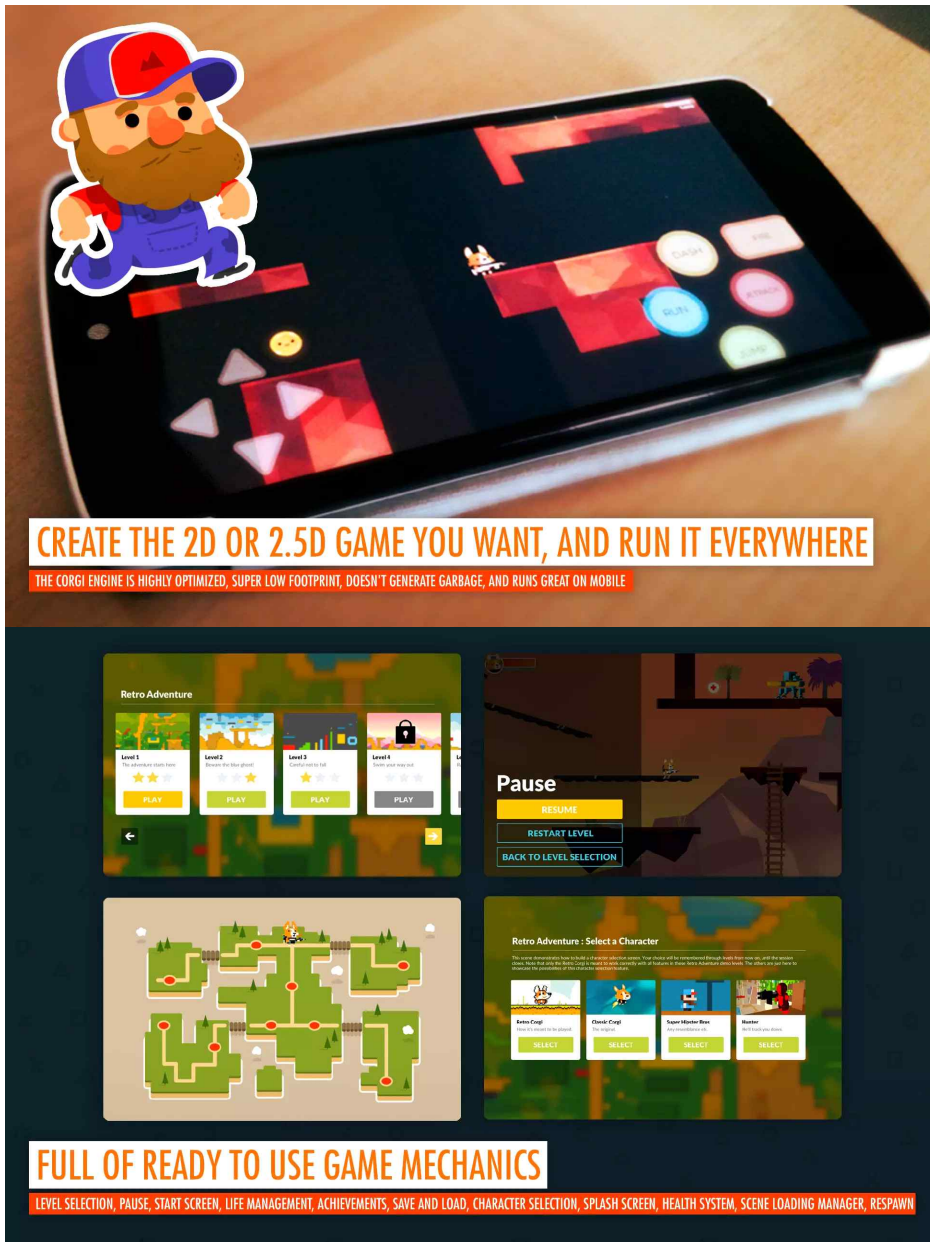
- 데모 Tilemap/Platform prefab을 열어 레벨 제작 시작. LevelManager 설정(스폰포인트, 체크포인트) 연결. ([링크](#))[7])

6.4 애니메이션 연동

- Ability가 제공하는 애니메이션 파라미터 이름(예: Dashing, Jumping 등)을 Animator에 맞춰 세팅. 문서에 표기된 파라미터 사용 권장. ([링크](#))[8])

6.5 튜닝

- 인스펙터에서 JumpHeight, DashDistance, Gravity 등 파라미터를 조정하며 “감”을 맞춤. 필요하다면 CharacterJumpOverride 같은 트리거존으로 구간별 워크플로우도 가능. ([링크](#))[9])



7. 빠른 코드 예제 — Ability 제어 (실무에서 자주 쓰는 패턴)

7.1 특정 Ability(예: 대시)를 코드로 즉시 시작시키기
Corgi API에 ‘CharacterDash’ 클래스가 있으며, 문서에 ‘InitiateDash()’ 같은 공개 메서드가 있습니다. (아래처럼 캐릭터에서 해당 컴포넌트를 찾아 호출하면 프로그래밍적으로 대시를 트리거할 수 있습니다).

```
using MoreMountains.CorgiEngine;
using UnityEngine;

public class DashTriggerExample : MonoBehaviour
{
    void Update()
    {
        if (Input.GetKeyDown(KeyCode.K)) // 예시: K 눌렀을 때 강제 대시
        {
            var dash = FindObjectOfType<Character>().GetComponent<CharacterDash>();
            if (dash != null && dash.DashAuthorized())
            {
                dash.InitiateDash(); // 문서화된 API 호출
            }
        }
    }
}
```

〉 참고: ‘CharacterDash’의 메서드 이름/조건은 공식 API 문서를 확인하세요. 위 예시는 문서에 기반한 전형적 사용 예입니다. ([링크](#))[10])

7.2 점프 파라미터를 트리거 존에서 오버라이드

‘CharacterJumpOverride’ 같은 컴포넌트로 특정 영역에서 점프 높이/점프 가능성 등을 자동으로 바꿀 수 있습니다. 이를 트리거 존에 붙이면, 캐릭터가 구역을 지날 때 자동으로 동작을 바꿉니다. ([링크](#))[9])

8. 레벨/디자인 팁

- 감(Feel) 튜닝: 점프 높이·중력·대시 거리 등은 플레이 감에 직접적 영향> 디자이너가 직접 조절 가능한 튜닝 변수로 노출하세요. ([링크](#))[11])
- 애니메이션 파라미터 일관성: Ability들이 사용하는 애니메이션 이름/파라미터가 문서화되어 있으니 Animator와 반드시 일치시키기. ([링크](#))[8])
- MMFeedbacks 활용: 공격·대시·피격 등에 즉각적인 피드백(카메라 흔들림, 파티클)을 연결하면 게임 피감(게임 feel)이 확 뜹니다. ([YouTube](#))[12])

9. 멀티플레이 / 플랫폼 / 호환성

- 로컬 멀티: 패키지 자체에서 로컬 멀티 샘플 제공(분할 입력 등). 네트워크 멀티플레이는 별도 네트워크 레이어(예: Mirror, Netcode)와의 연동 작업 필요. ([링크](#))[1])
- 모바일: 터치 입력·가상 조이스틱 샘플이 포함되어 있어 모바일 포팅이 비교적 수월. ([링크](#))[1])

10. 퍼포먼스 & 유지보수 포인트

- Ability가 많은 캐릭터는 업데이트 비용이 커질 수 있으니, 사용하지 않는 Ability는 비활성화(Enable=false) 유지.
- 데모/샘플 에셋이 많아 프로젝트 용량이 커짐> 릴리즈 빌드 시 불필요한 데모 에셋 제외 권장. ([링크](#))[7])

11. 확장성 & 개발자 친화성

- 소스 API & 문서: 공식 문서와 API 레퍼런스가 잘 정리되어 있어 Ability를 상속해 커스텀 동작을 만들기 쉬움(예: CharacterDamageDash처럼 기본 클래스를 확장). ([링크](#))[13])
- 커뮤니티(Discord)·포럼·유튜브 튜토리얼도 활발 → 특정 기능 확장·문제해결에 도움됨. ([링크](#))[14])

12. 참고 리소스 (핵심)

- 공식 소개 및 데모: Corgi Engine 홈페이지. ([링크](#))[7])
- 공식 문서 / API 레퍼런스(Ability 클래스 목록, 함수 등). ([링크](#))[4])
- Asset Store(구매/버전/릴리즈 노트). ([Unity Asset Store](#))[2])
- 설치/레시피 가이드(빈 프로젝트에 импорт 권장 등). ([링크](#))[3])

13. 마무리(요약)

Corgi Engine은 2D/2.5D 플랫폼 게임의 ‘움직임·레벨·데모’ 대부분을 바로 제공해주는 상용 툴킷입니다. 빠른 프로토타입 제작과 완성도 높은 컨트롤을 원한다면 매우 효율적이며, 문서·API·데모가 충실해서 팀 도입 시 학습곡선 또한 완만합니다. 다만 패키지 크기와 기본 샘플이 많으니 프로젝트 구조·에셋 관리 정책을 초기부터 세워두는 게 중요합니다. ([링크](#))[1])