



[4주차] 2D RunGame-1

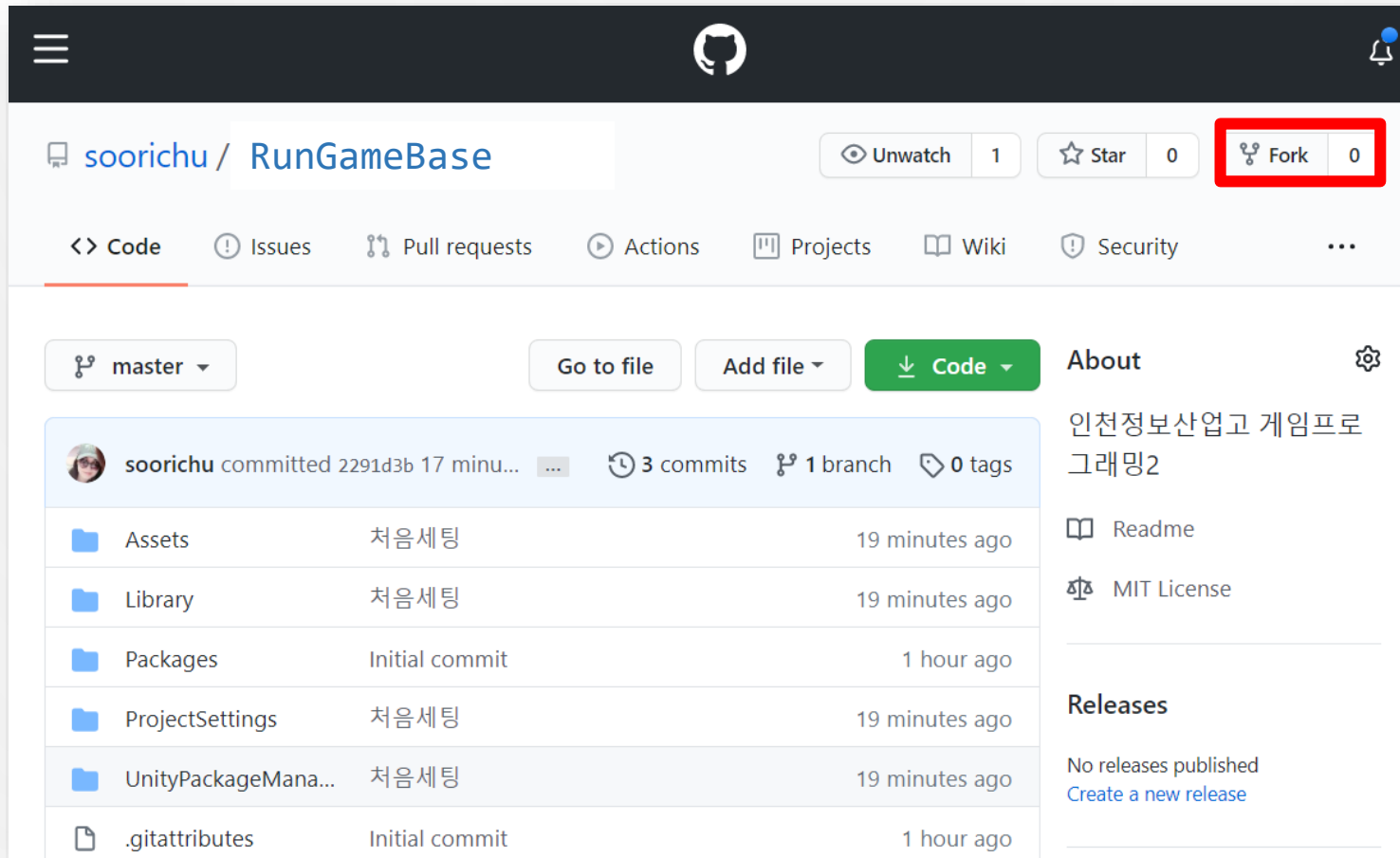
인천정보과학고등학교 3학년 전산과 게임프로그래밍2

© all right reserved 인천정보과학고 IT소프트웨어과 정수영

깃허브 포크하기

깃허브 포크

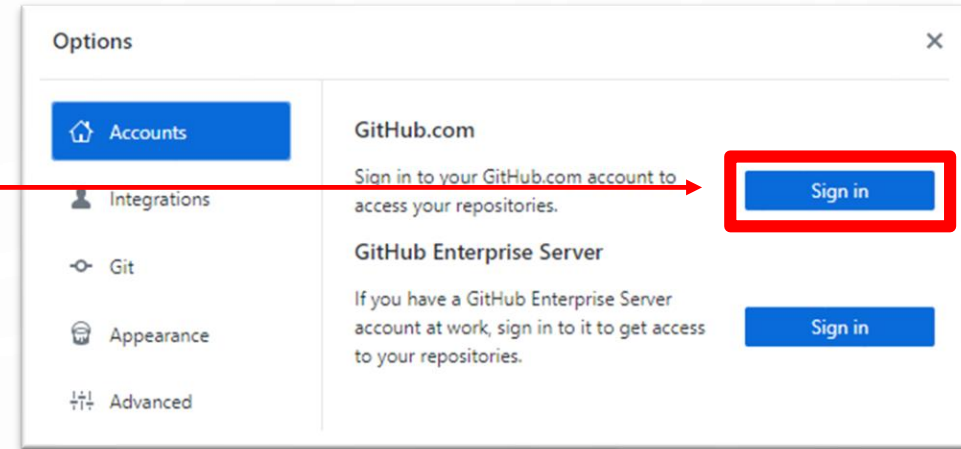
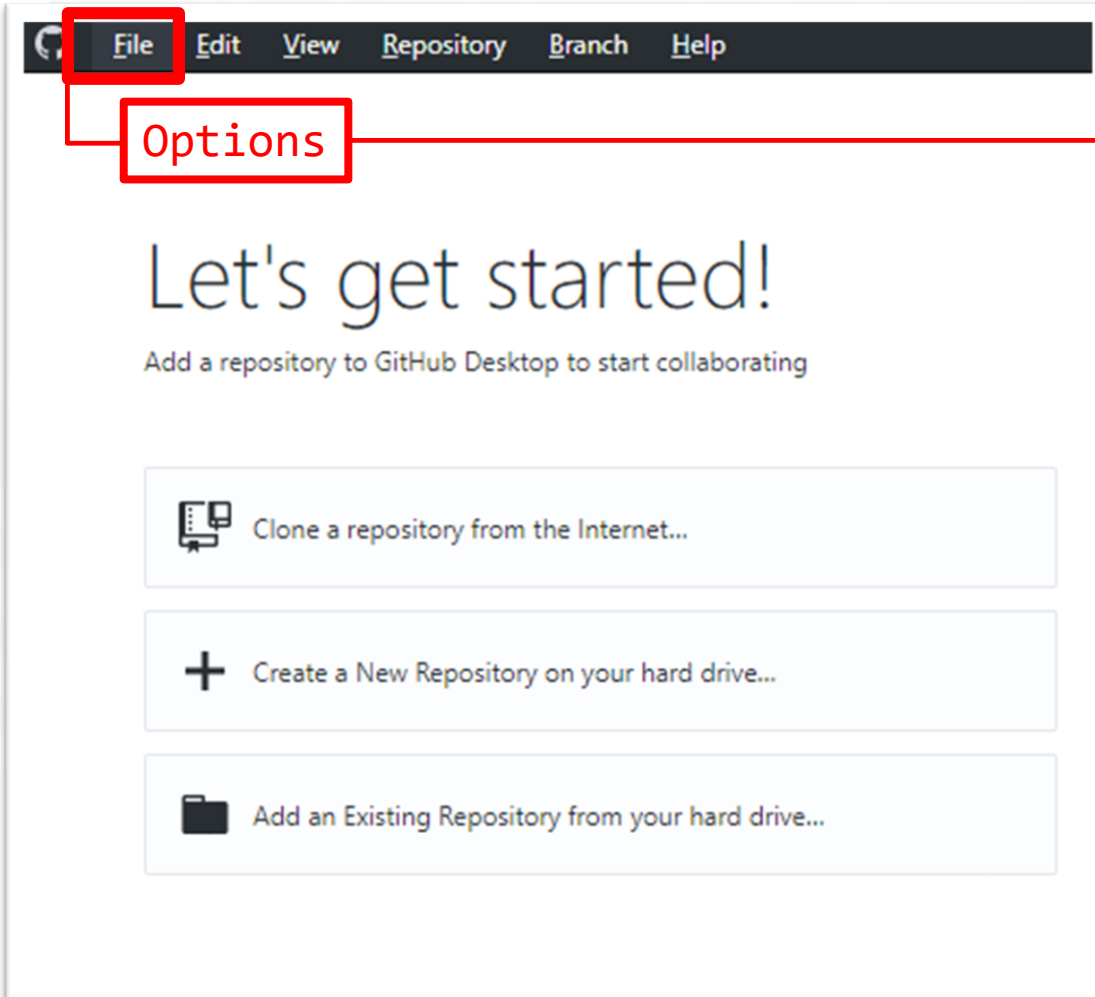
- <https://github.com/soorichu/RunGameBase> > Fork 클릭



Git Desktop Login



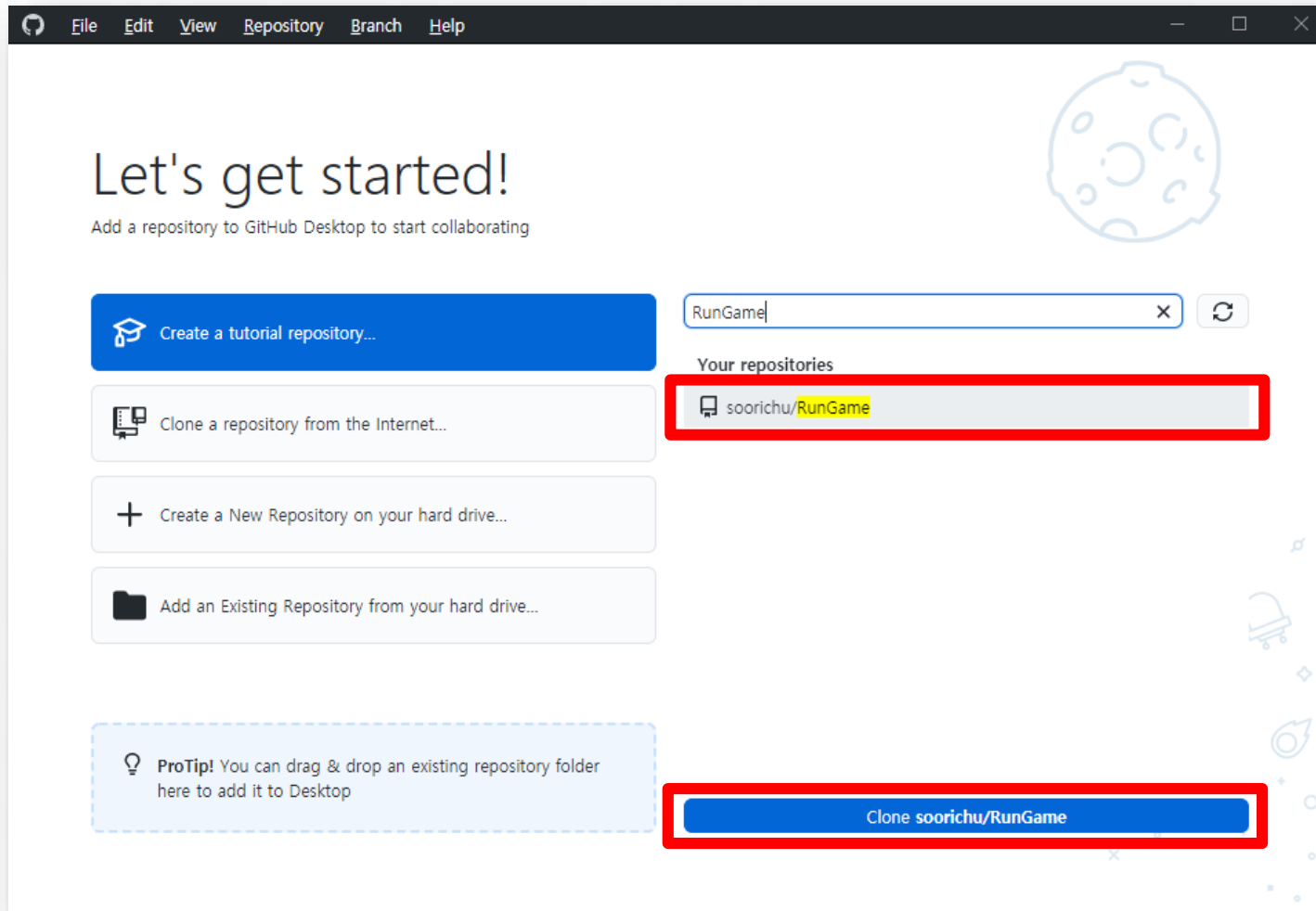
Github Desktop



깃헙 로그인 진행
크롬에서 다음과 같은 메시지가 나오면



Clone Repository



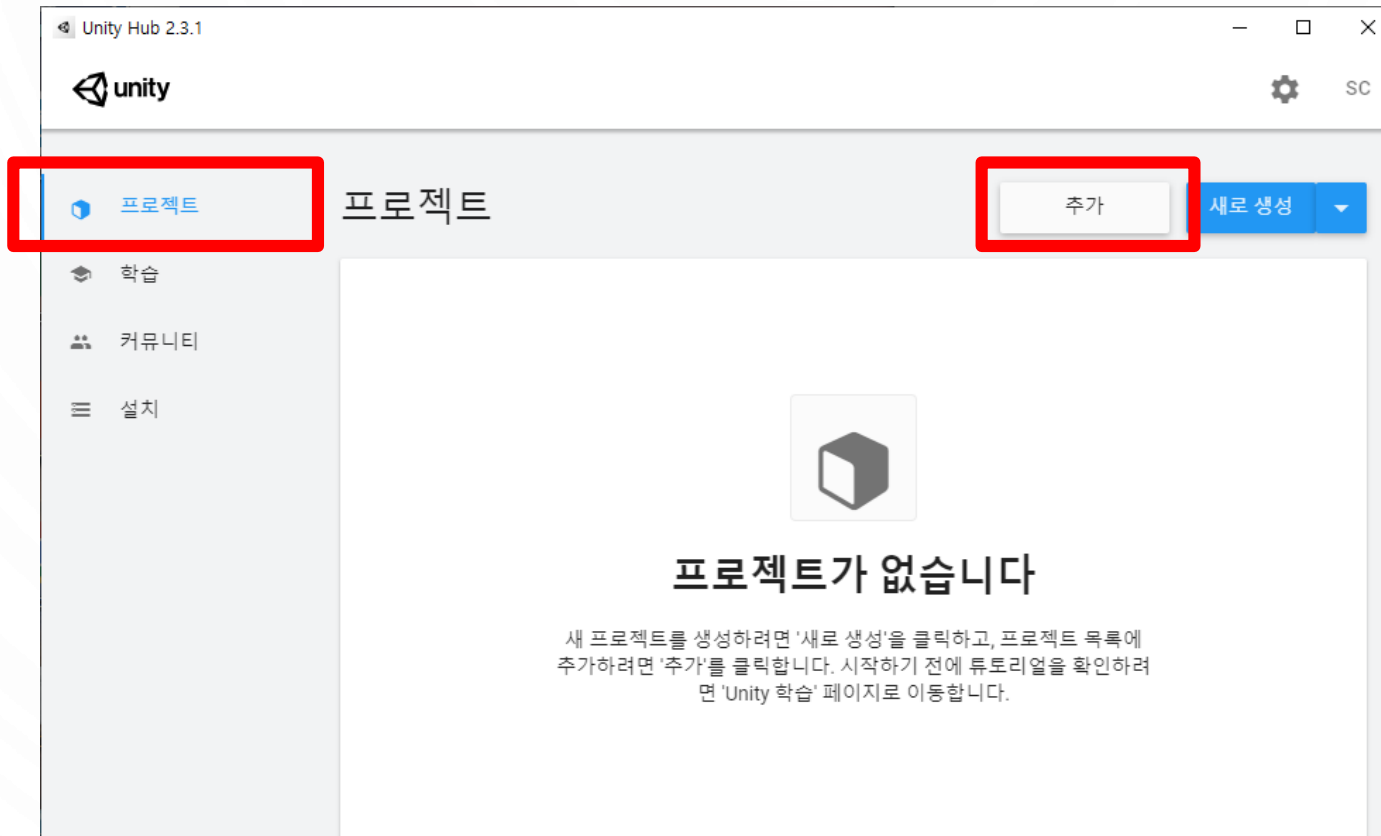
<user>/RunGameBase 선택

Clone 클릭

Project Open



- Unity Hub에서 프로젝트 > 추가 > RunGame 폴더 선택



발판과 데드존

시작지점 발판 만들기

① Platform_Long 스프라이트를
하이러키 창으로 드래그&드롭

② 게임 오브젝트의 이름을 Start Platform으로 변경

③ 위치를 (0, -1, 0)으로 변경

④ Box Collider 2D 컴포넌트 추가(Add Component > Physics 2D > Box Collider 2D)

▶ 시작 지점 발판 만들기

데드존 만들기

① 빈 게임 오브젝트 생성

② 이름은 Deadzone, 태그는 Dead, 위치는 (0, -8, 0)으로 변경

③ Box Collider 2D 컴포넌트 추가(Add Component > Physics 2D > Box Collider 2D)

④ Box Collider 2D 컴포넌트의 Is Trigger 체크, Size는 (50, 2)로 변경

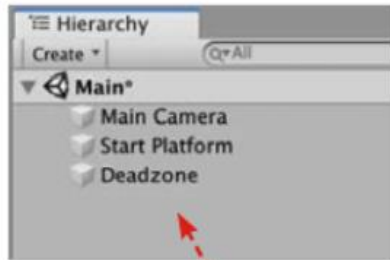
The image shows the Unity Hierarchy and Inspector panels. The Hierarchy panel on the left shows the 'Create' menu with 'Create Empty' selected. The Inspector panel on the right shows the 'Deadzone' object with the following properties:

- Name: Deadzone
- Tag: Dead
- Layer: Default
- Transform: Position (X: 0, Y: -8, Z: 0), Rotation (X: 0, Y: 0, Z: 0), Scale (X: 1, Y: 1, Z: 1)
- Box Collider 2D: Material: None (Physics Material 2D), Is Trigger: checked, Used By Effector: unchecked, Used By Composite: unchecked, Auto Tiling: unchecked, Offset (X: 0, Y: 0), Size (X: 50, Y: 2), Edge Radius: 0

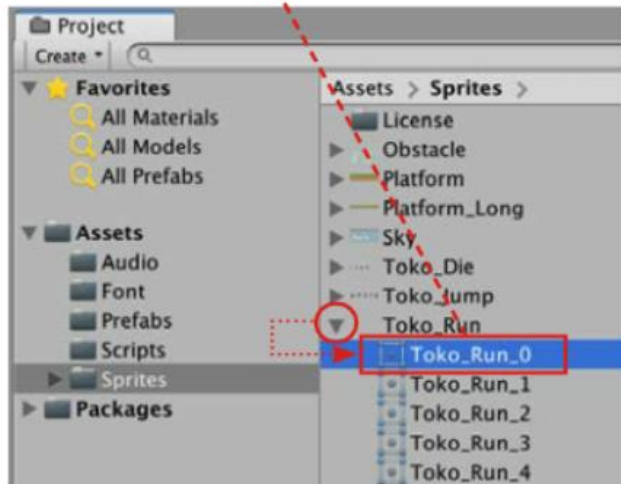
▶ 데드존 만들기

플레이어 준비하기

Player 게임 오브젝트 구성

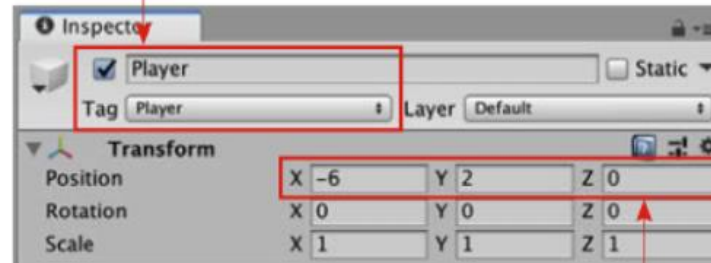


① Toko_Run_0 스프라이트를 하이어라키 창으로 드래그&드롭

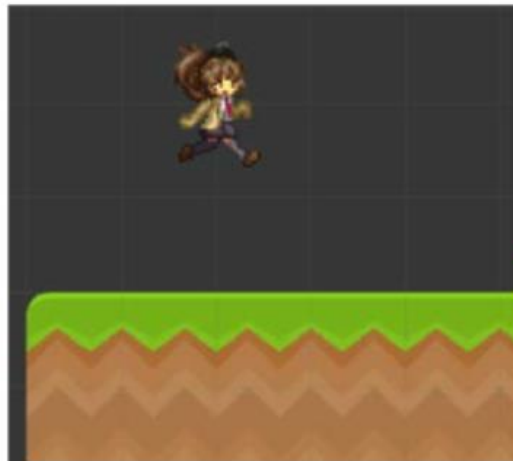


▶ 플레이어 게임 오브젝트 생성

② 이름은 Player, 태그는 Player로 변경

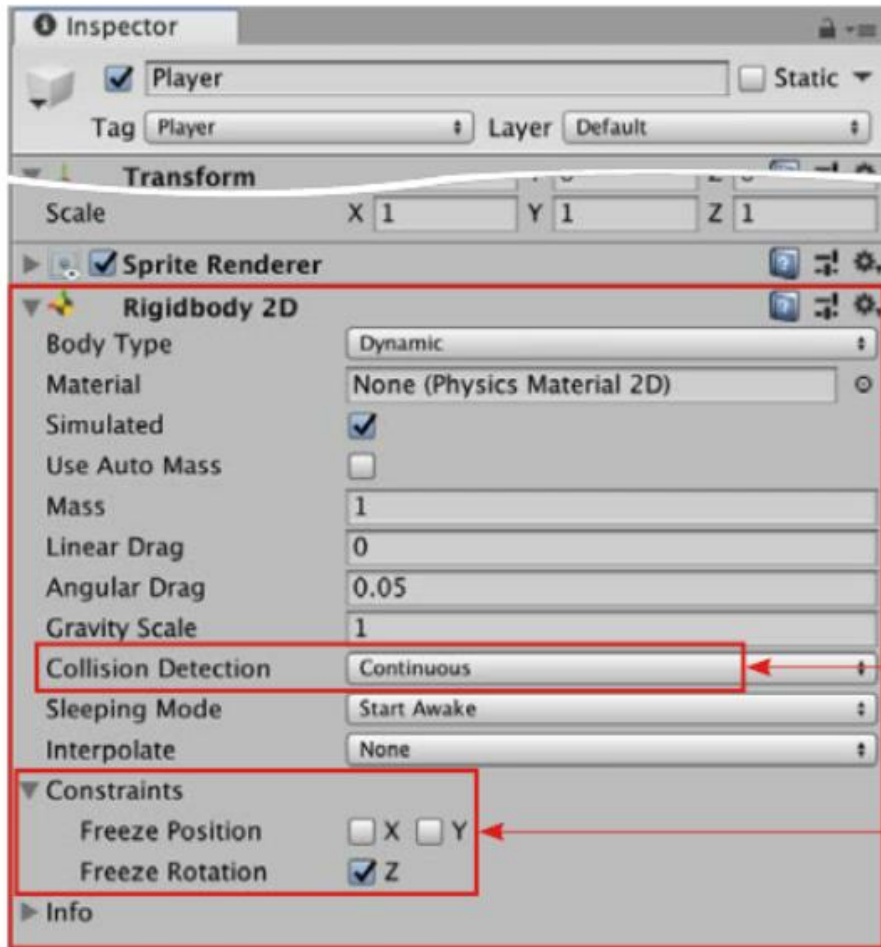


③ 위치를 (-6, 2, 0)으로 변경



배치된 Player 게임 오브젝트의 모습

Rigidbody 2D 컴포넌트 추가



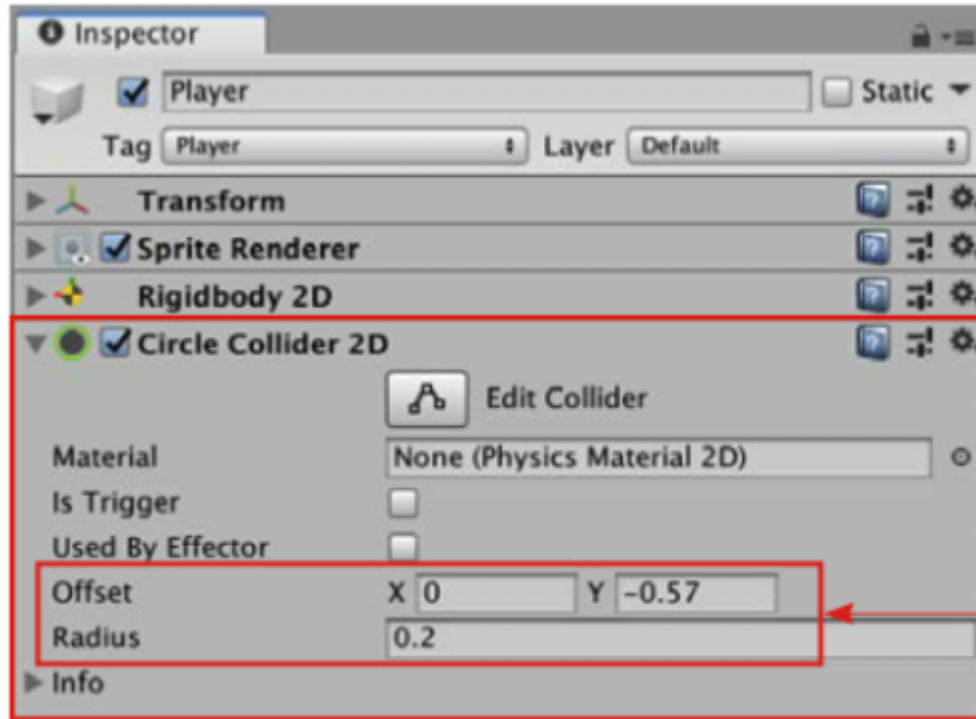
① Player 게임 오브젝트에 Rigidbody 2D 컴포넌트 추가(Add Component > Physics 2D > Rigidbody 2D)

② Collision Detection을 Continuous로 변경

③ Constraints 펼치기 > Freeze Rotation의 Z 체크

▶ 리지드바디 2D 컴포넌트 추가

Collider 2D 컴포넌트 추가

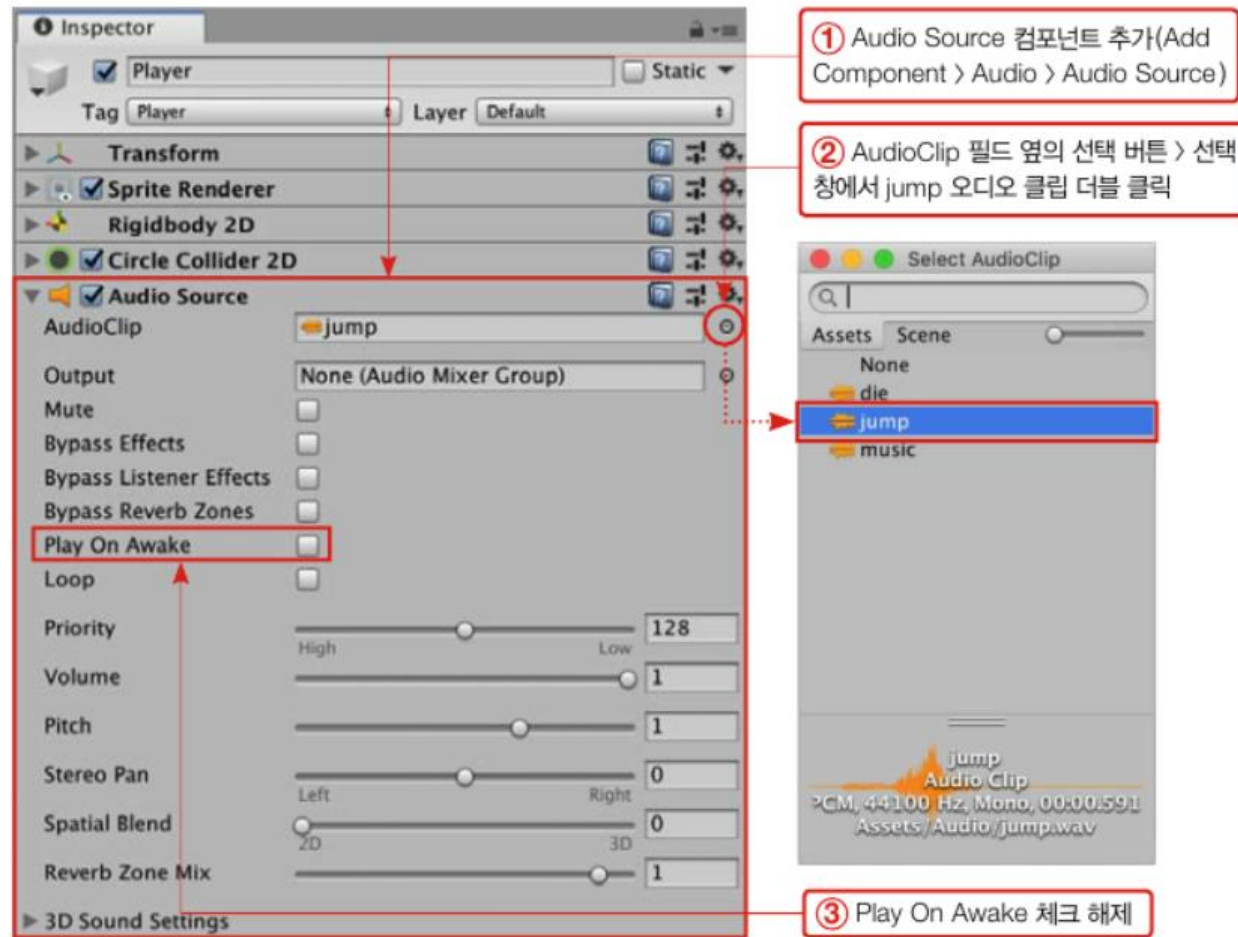


① Circle Collider 2D 추가 (Add Component > Physics 2D > Circle Collider 2D)

② Offset을 (0, -0.57), Radius을 0.2로 변경

▶ 써클 콜라이더 2D 컴포넌트 추가

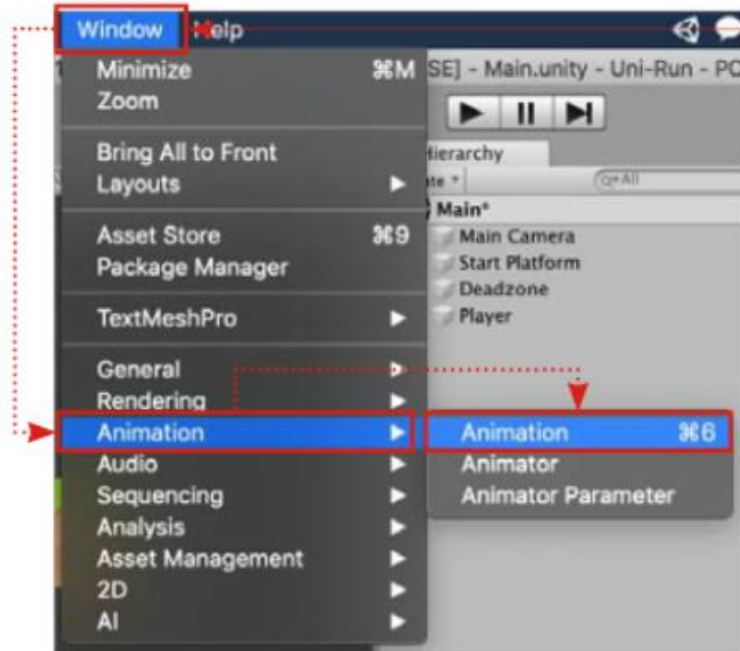
Audio Source 컴포넌트 추가



▶ 오디오 소스 컴포넌트 추가

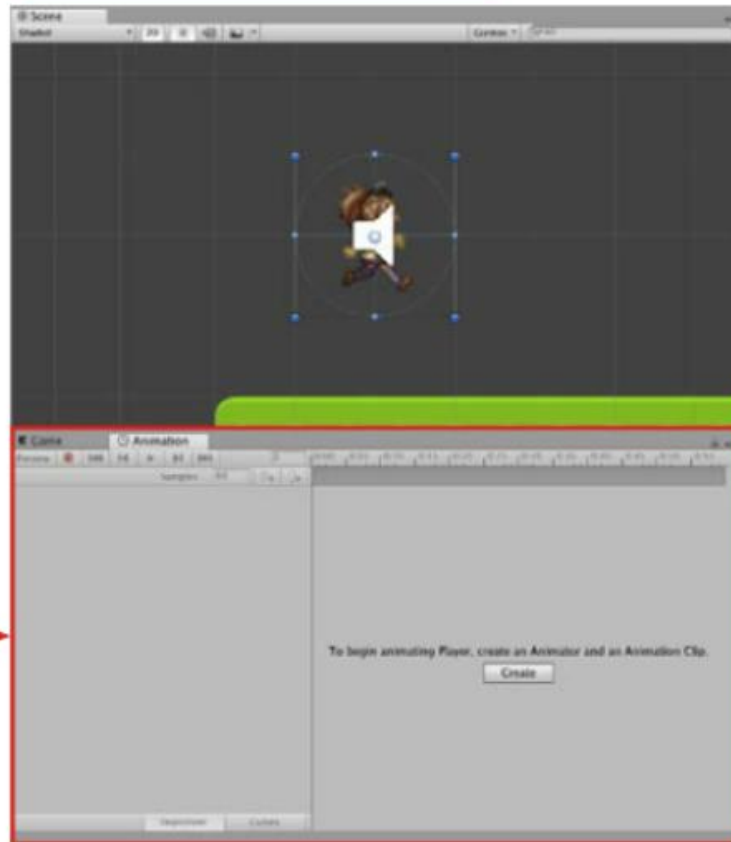
애니메이션 만들기

애니메이션 만들기



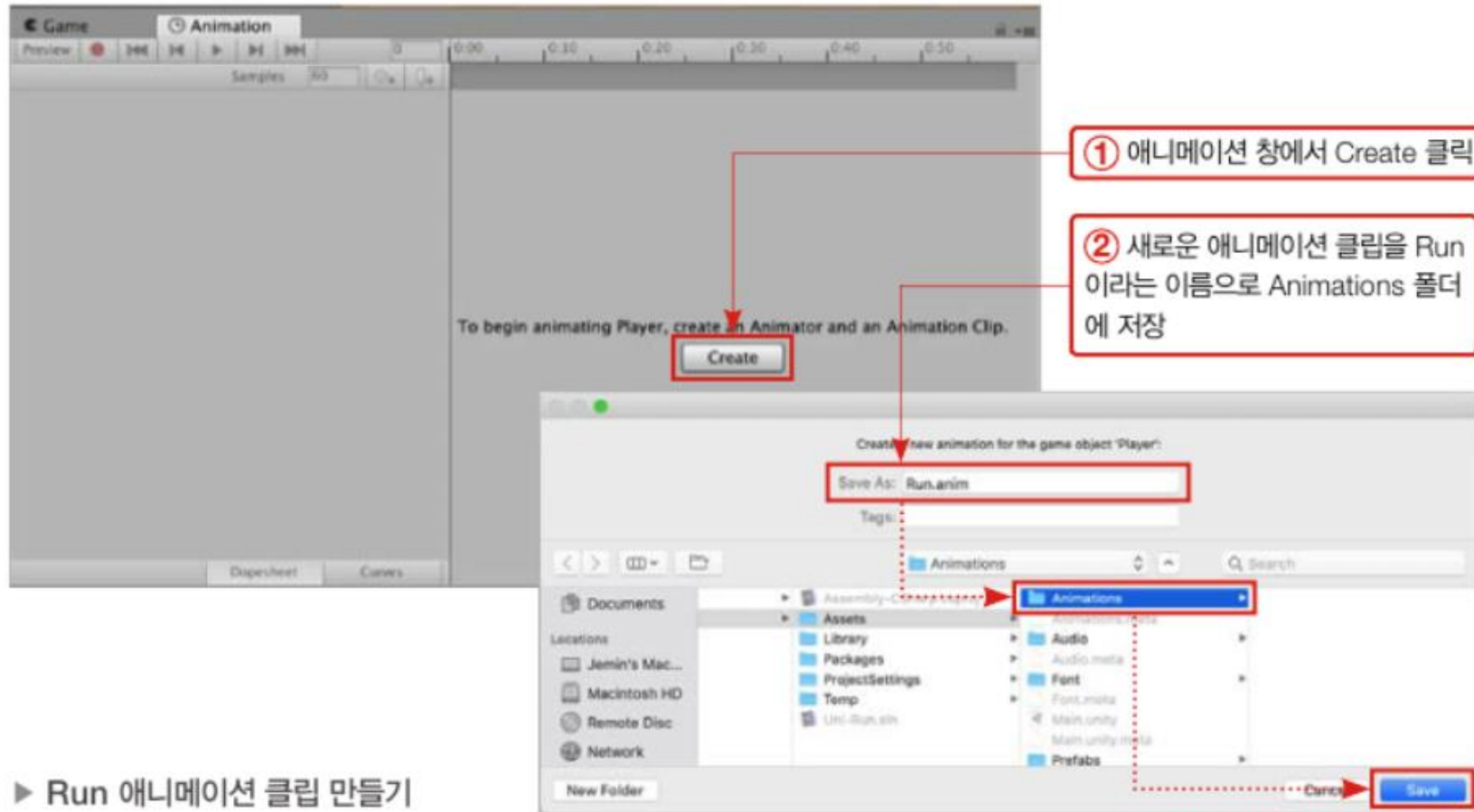
① 상단 메뉴의 Window > Animation > Animation 클릭

② 애니메이션 창을 드래그하여 적절한 곳에 배치



▶ 애니메이션 창 열기

Run 애니메이션 클립(1)



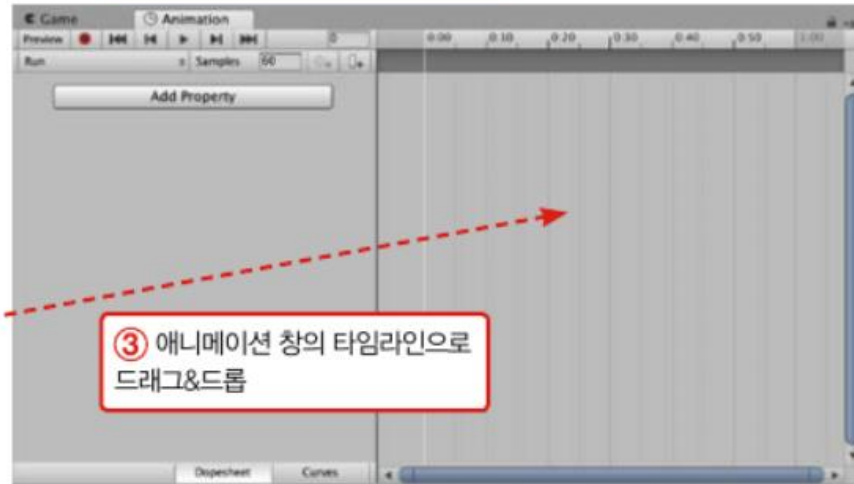
▶ Run 애니메이션 클립 만들기

Run 애니메이션 클립(2)

① 프로젝트 창에서 Sprites 폴더로 이동

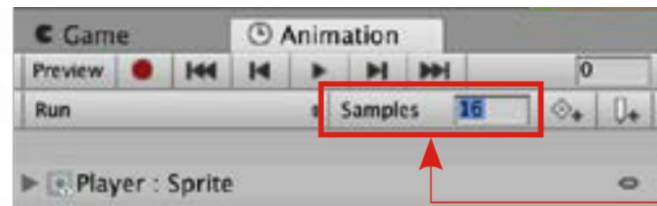


② Toko_Run 스프라이트 펼치기 >
[Shift+클릭]으로 모두 선택



③ 애니메이션 창의 타임라인으로
드래그&드롭

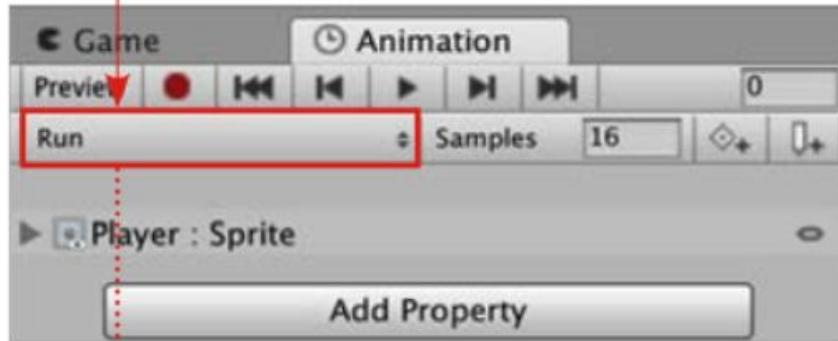
▶ Run 애니메이션 클립 구성하기



④ Samples 16으로 변경

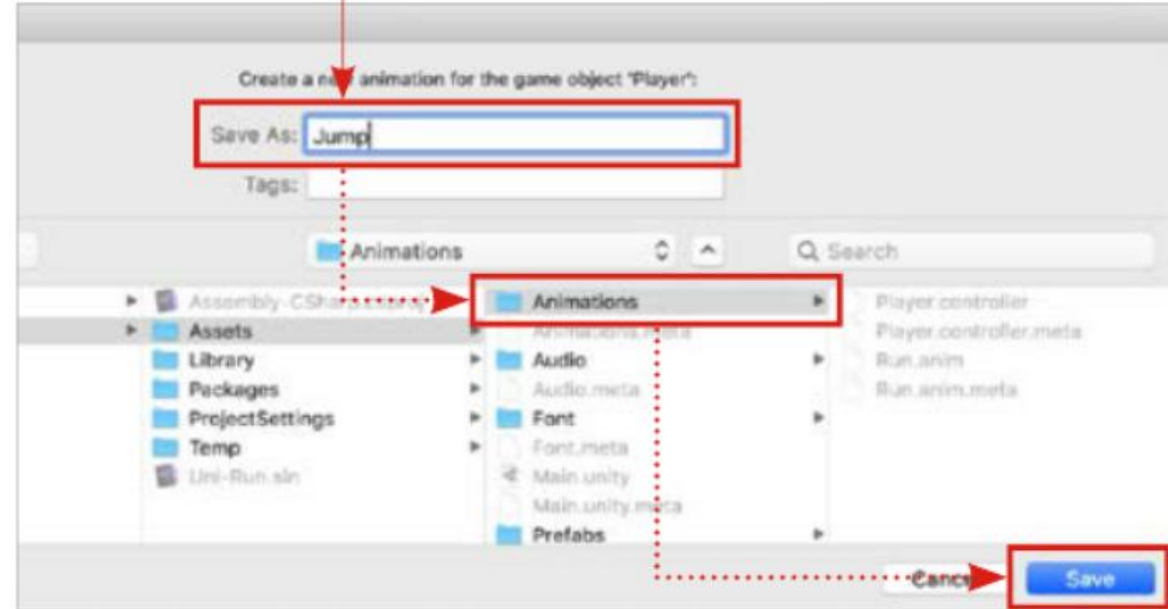
Jump 애니메이션 클립(1)

① 애니메이션 창에서 Run 클릭 >
Create New Clip... 클릭

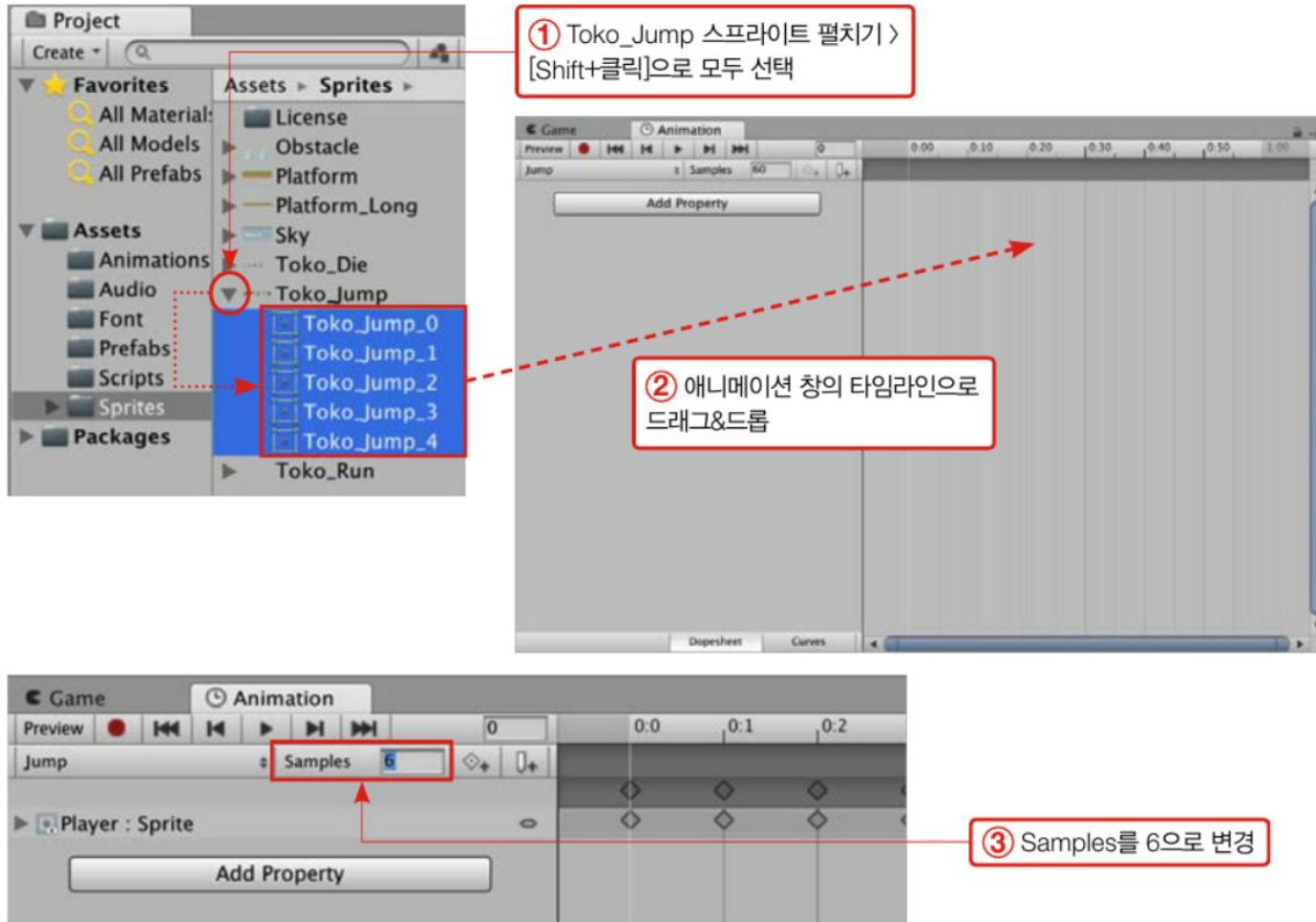


▶ Jump 애니메이션 클립 만들기

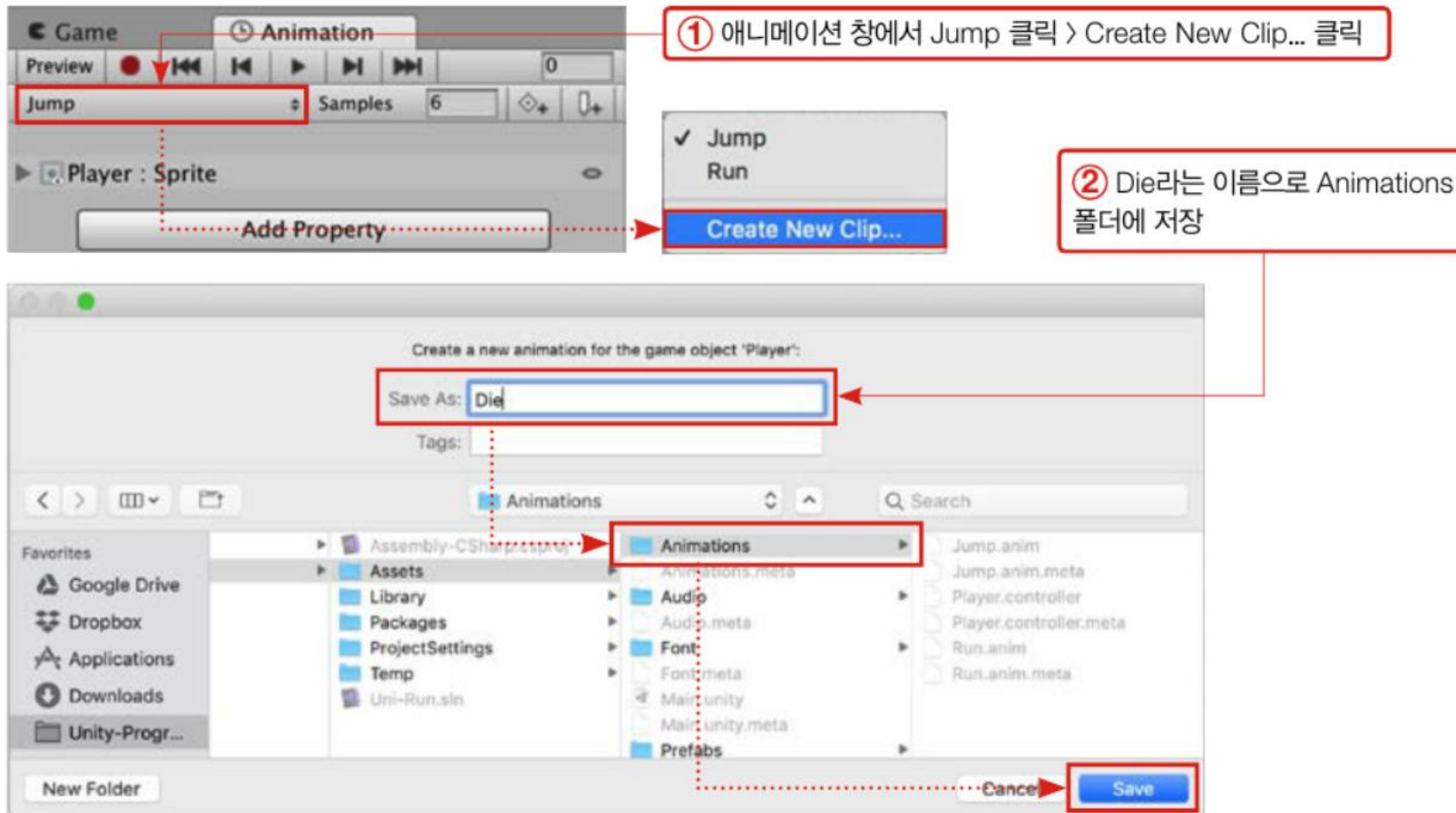
② Jump라는 이름으로 Animations 폴더에 저장



Jump 애니메이션 클립(2)

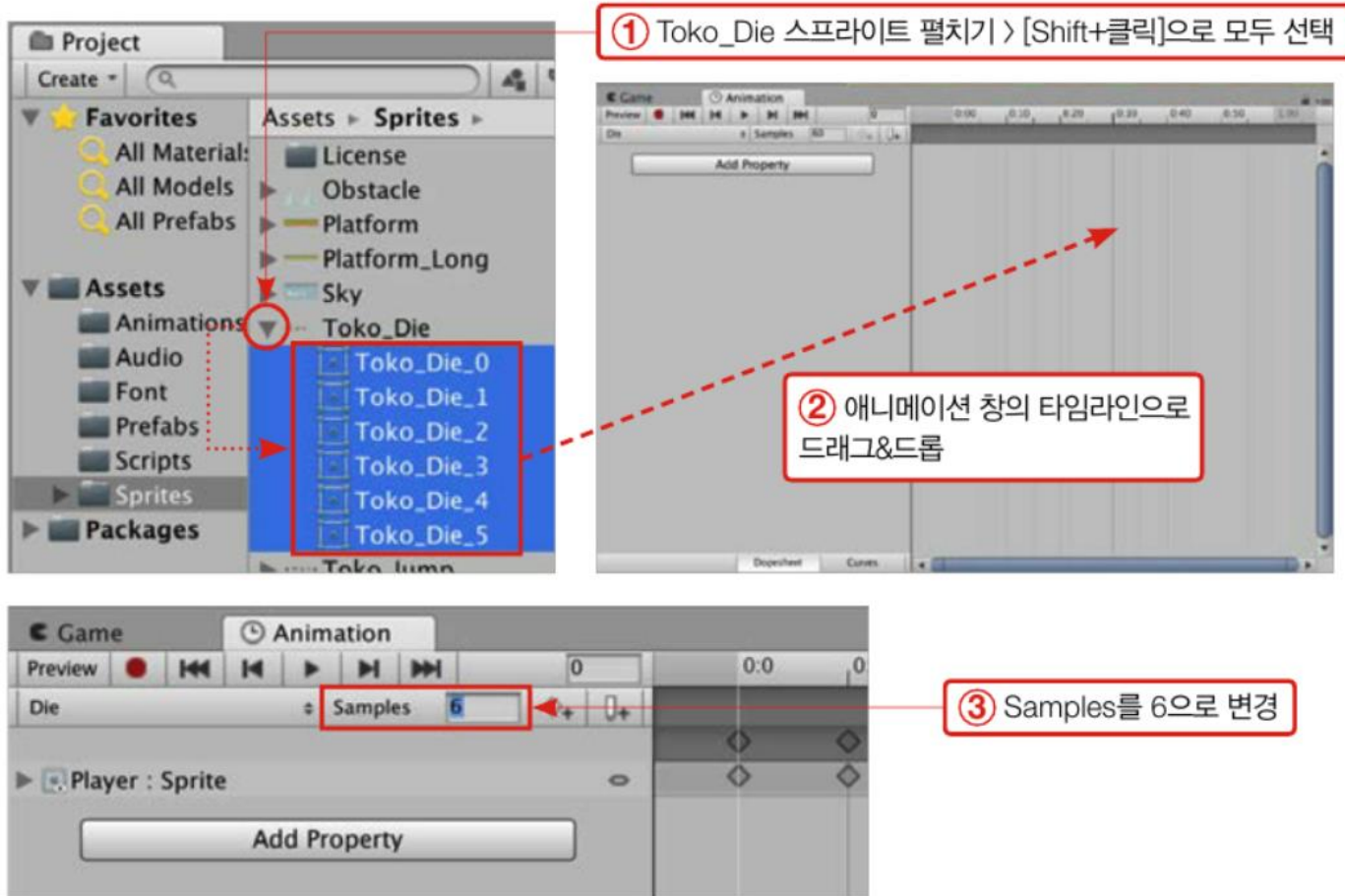


Die 애니메이션 클립(1)



▶ Die 애니메이션 클립 만들기

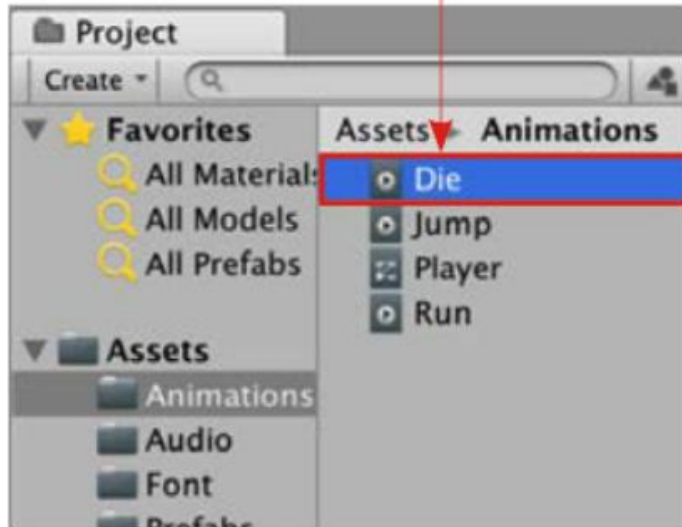
Die 애니메이션 클립(2)



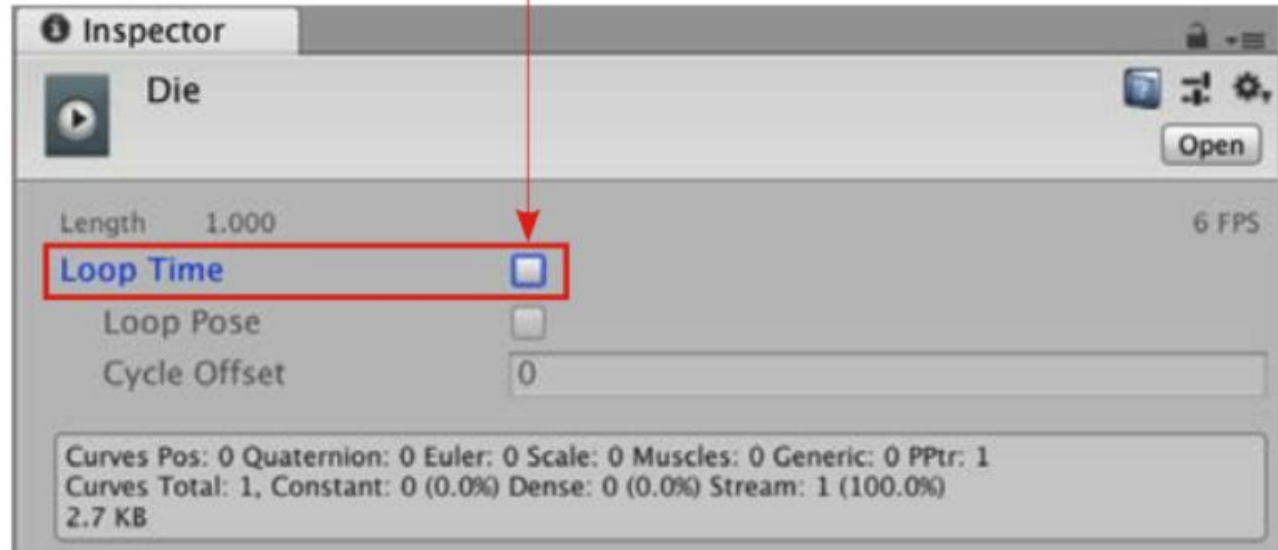
▶ Die 애니메이션 클립 구성하기

Die 애니메이션 클립(3)

① Animations 폴더로 이동 >
Die 애니메이션 클립 선택



② 인스펙터 창에서 Loop Time
체크 해제



▶ Die 애니메이션 클립의 루프 설정 해제

애니메이터

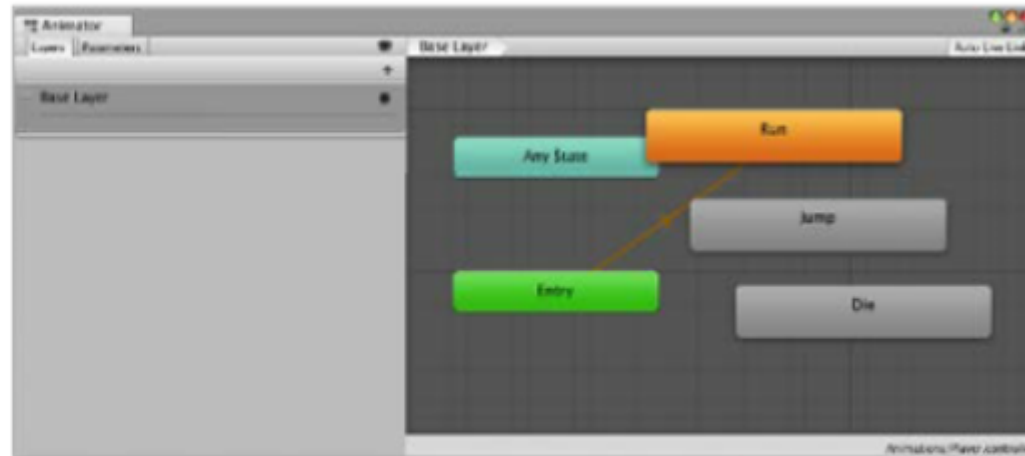
애니메이터 창 열기

① Player를 선택한 상태에서 상단에 Windows 메뉴 선택



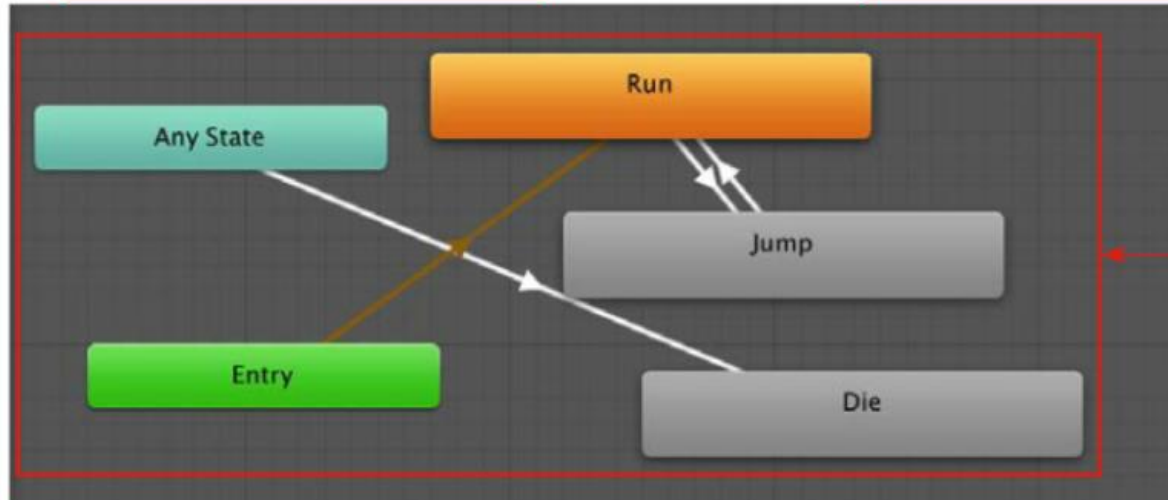
▶ 애니메이터 창 열기

② 상단 메뉴에서 Window > Animation > Animator 클릭



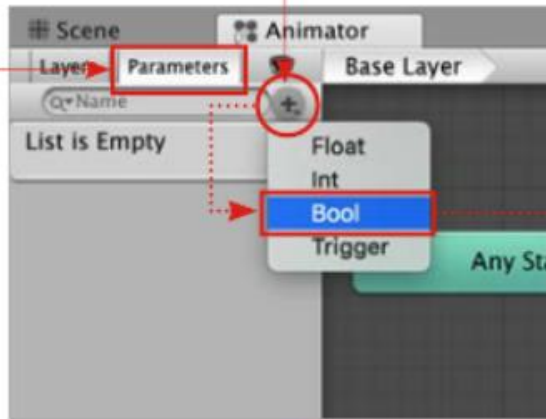
③ 열린 애니메이터 창을 적절한 위치로 드래그하여 배치

전이 화살표 구성하기

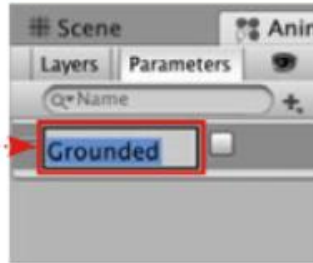


파라미터 추가

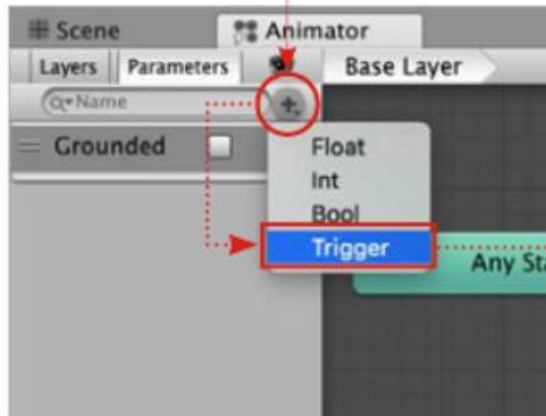
① 애니메이터 창에서 Parameters 탭 클릭



② + 버튼 클릭 > Bool 클릭 > 생성된 파라미터 이름을 Grounded로 변경

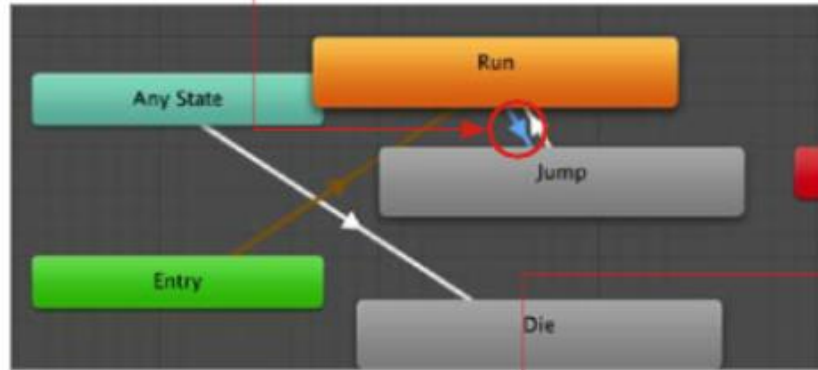


③ + 버튼 클릭 > Trigger 클릭 > 생성된 파라미터 이름을 Die로 변경



Run → Jump 전이 설정

① Run → Jump 전이(화살표) 클릭

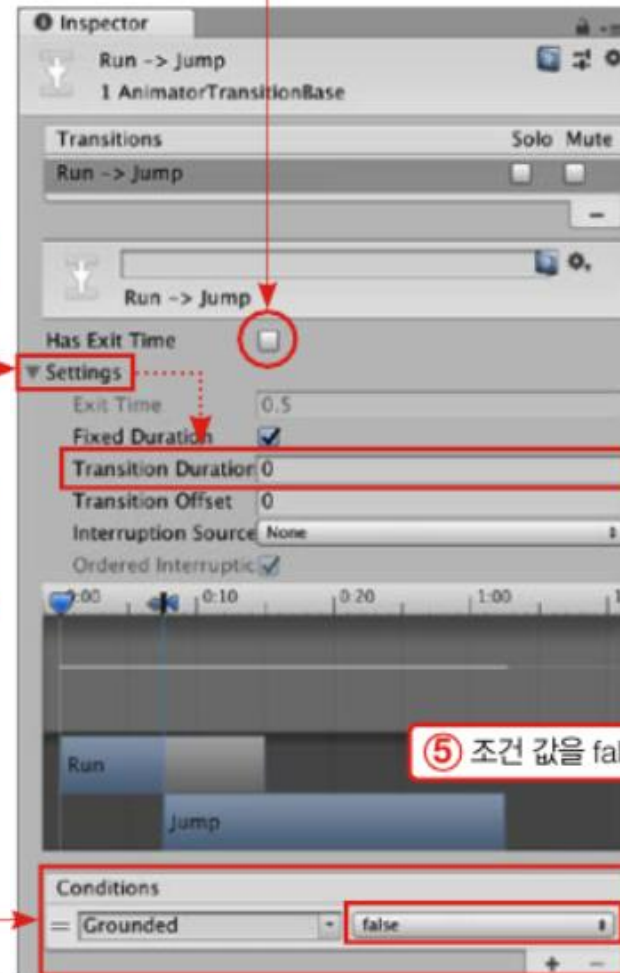


③ Settings 탭 펼치기 > Transition Duration을 0으로 변경

④ 조건에 Grounded 추가(Conditions의 + 버튼 클릭)

▶ Run → Jump 전이 설정하기

② Has Exit Time 체크 해제



⑤ 조건 값을 false로 변경

Jump → Run 전이 설정

① Jump → Run 전이 클릭

② Has Exit Time 체크 해제

③ Settings 탭 펼치기 > Transition Duration을 0으로 변경

④ 조건에 Grounded 추가 (Conditions의 + 버튼 클릭)

⑤ Grounded의 조건값을 true로 변경

▶ Jump → Run 전이 설정하기

The image shows the Unity Animator interface. On the left, the 'Base Layer' state machine is visible with states 'Any State' (teal), 'Run' (orange), 'Jump' (grey), 'Entry' (green), and 'Die' (grey). A transition from 'Any State' to 'Run' is highlighted with a red circle and an arrow pointing to it from label ①. On the right, the 'Inspector' panel shows the 'Jump -> Run' transition settings. Label ② points to the 'Has Exit Time' checkbox, which is unchecked. Label ③ points to the 'Settings' dropdown menu. Label ④ points to the '+' button in the 'Conditions' section. Label ⑤ points to the 'true' value in the 'Conditions' dropdown. The 'Transition Duration' is set to 0. The 'Conditions' section shows 'Grounded' set to 'true'.

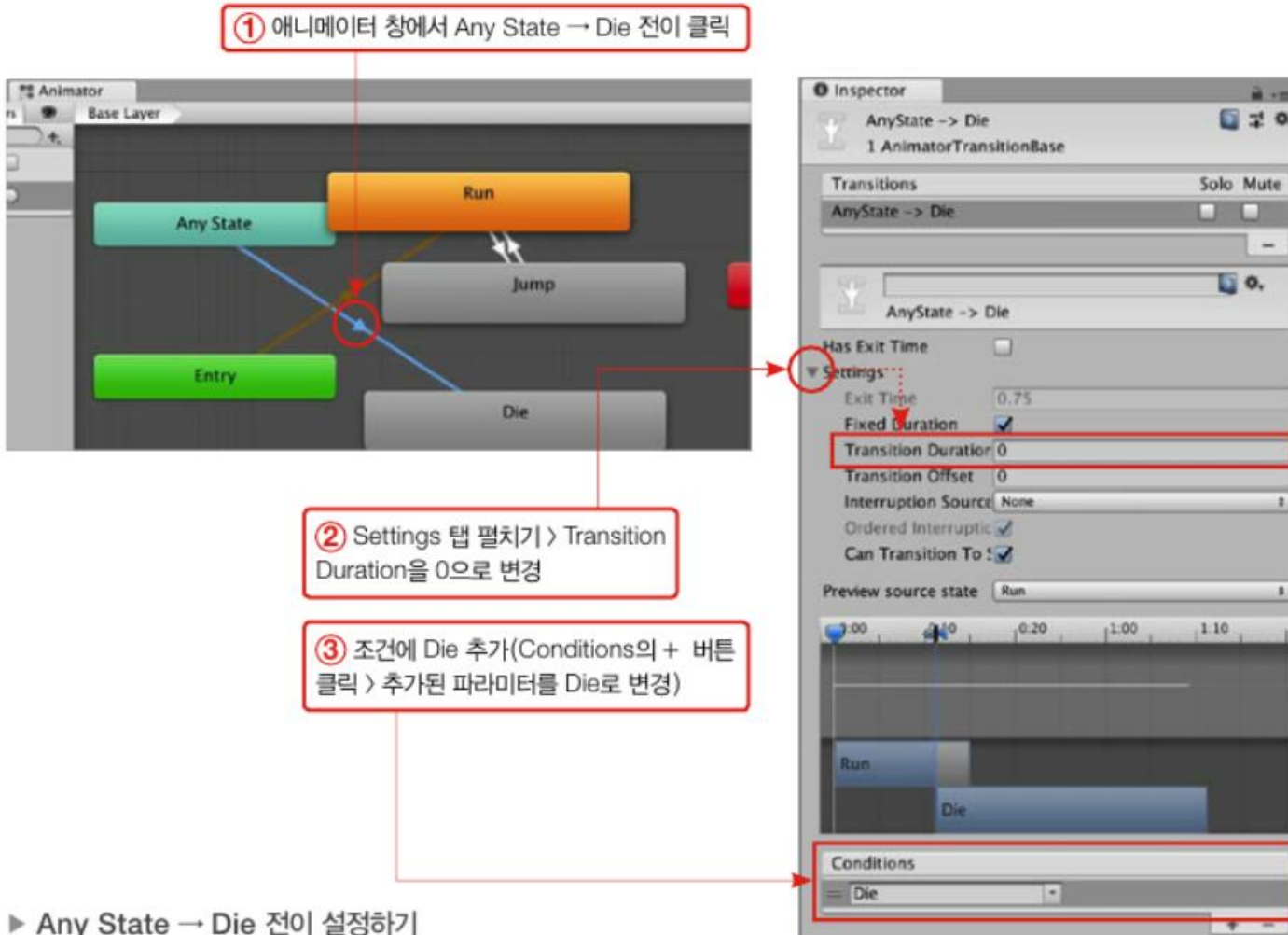
Any State → Die 설정

① 애니메이터 창에서 Any State → Die 전이 클릭

② Settings 탭 펼치기 > Transition Duration을 0으로 변경

③ 조건에 Die 추가(Conditions의 + 버튼 클릭 > 추가된 파라미터를 Die로 변경)

▶ Any State → Die 전이 설정하기

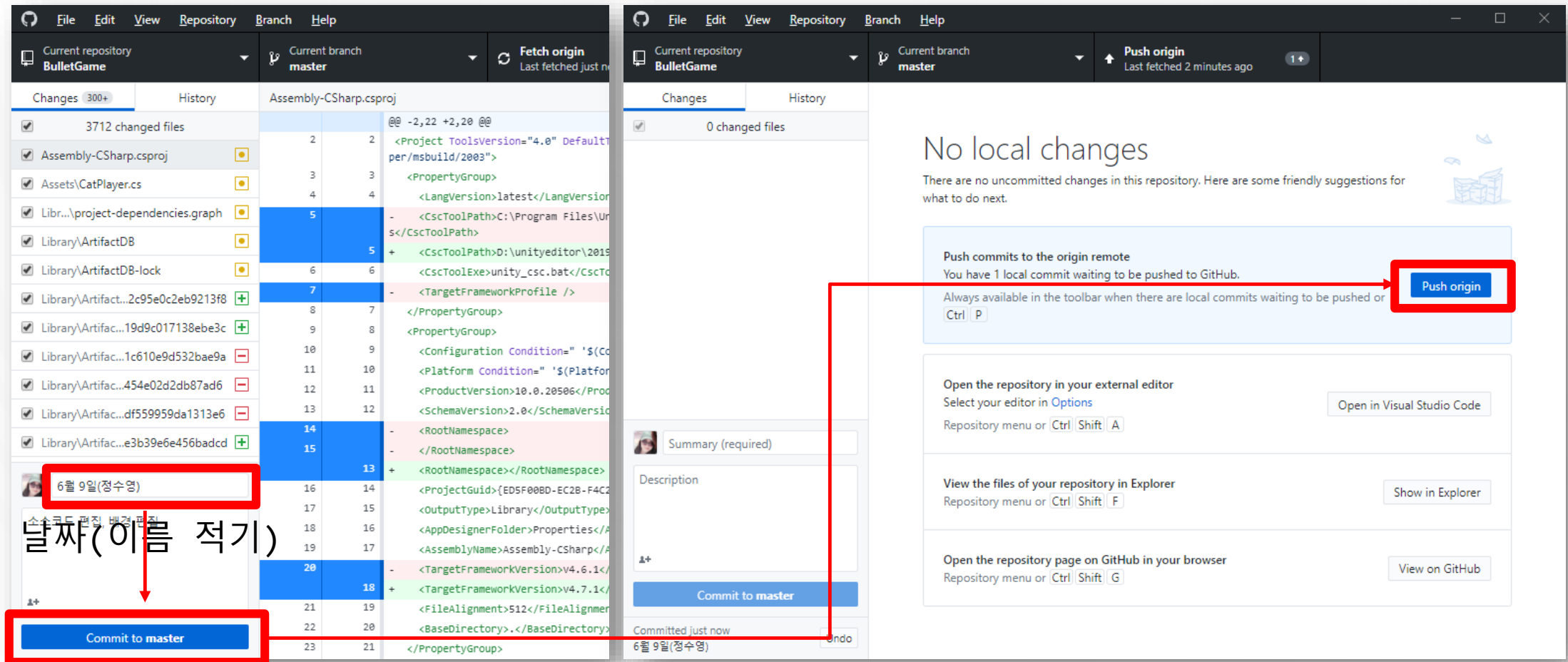


The image shows the Unity Animator interface. On the left, the Animator window displays a state machine with states: Any State (green), Run (orange), Entry (green), Jump (grey), and Die (grey). A transition arrow points from Any State to Die. A red circle highlights this transition, with an arrow pointing to the Inspector window on the right. The Inspector window shows the settings for the 'AnyState -> Die' transition. The 'Settings' tab is selected, and the 'Transition Duration' is set to 0. The 'Conditions' section at the bottom shows a dropdown menu with 'Die' selected. Red boxes and arrows highlight these specific settings.

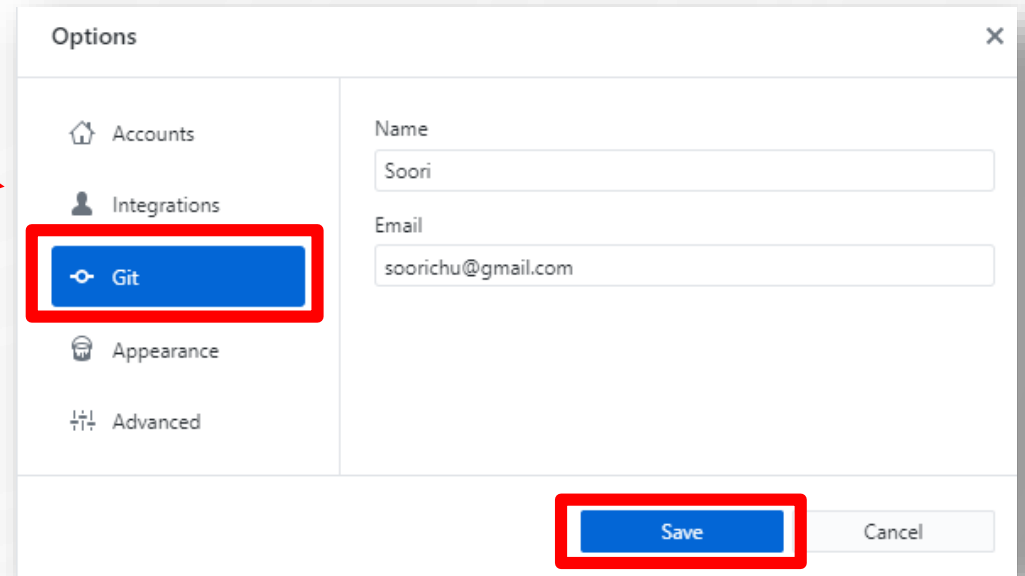
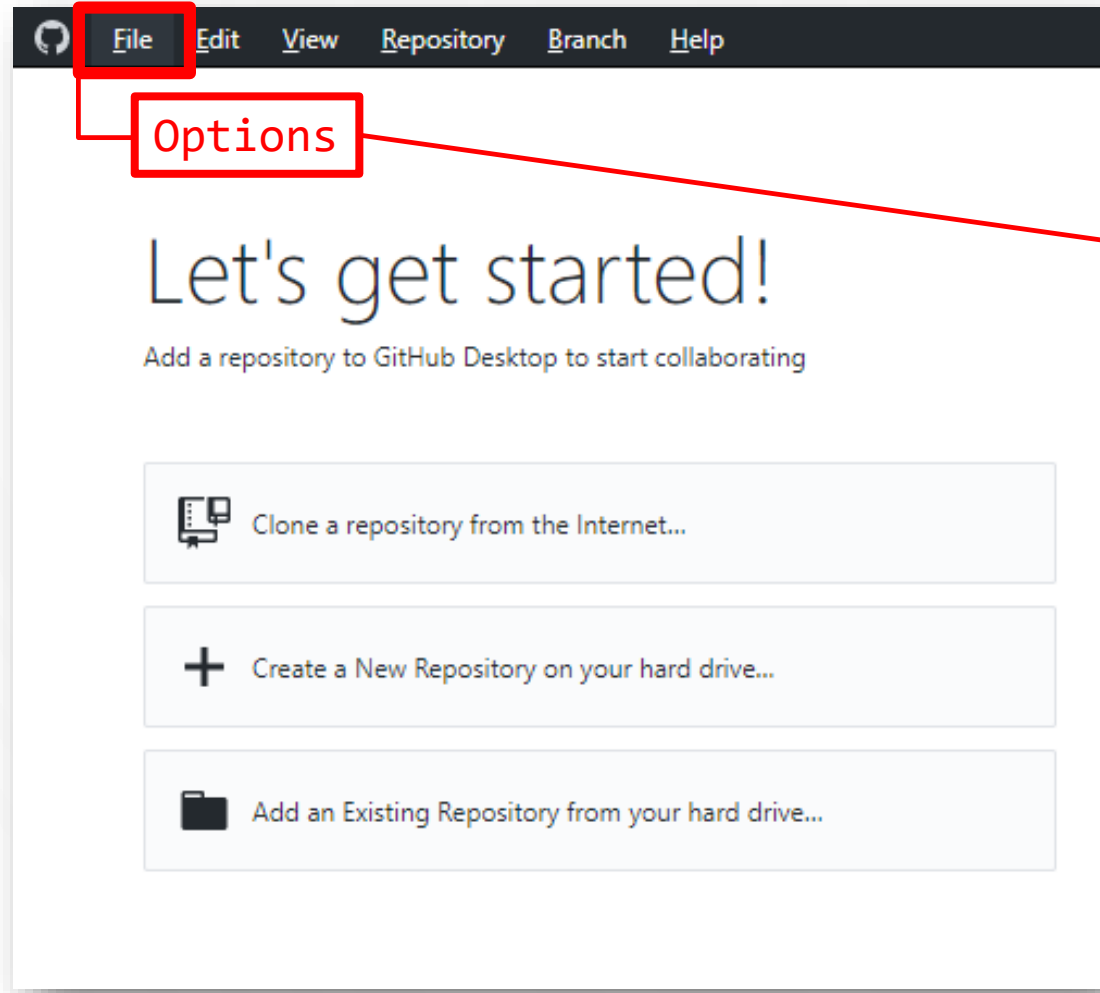
GITHUB PUSH

깃허브 올리기

- Unity 프로젝트 저장 후 모두 닫고 Git Desktop 열기



에러 날 때!



그냥 Name과 Email이 잘 채워진 것만 확인하고 Save 버튼 누르세요.
이제 다시 Commit > Push 시도해보세요. ^^

END

참고자료 : 레트로의 유니티 게임 프로그래밍 에센스