

## 대화 시스템

대화 시스템은 대화의 연출방법을 정의해놓은 '시나리오'와 대화의 내용들을 기록해 놓은 '스크립트'가 있습니다. 이때 스크립트는 스크립트 언어와는 달리 대본이라는 의미로 사용되는 것이므로 둘을 혼동하지 않도록 주의해주시요.

시나리오는 문자열의 형태로 된 명령어들의 집합으로 이루어져 있습니다. 유니티 런타임에 이 문자열들을 직접 해석하는 것은 성능상 좋지 않기에 이 문자열들을 바이트형태로 변환해 (바이트코드로 만들어) 사용합니다.

그래서 실질적으로 시나리오의 바이트 코드와 스크립트가 데이터로 사용됩니다. 그리고 이 데이터들은 유니티의 에셋으로 관리되며 어드레서블로 이 에셋들을 받아와야 합니다.

### 스크립트를 작성하는 방법

스크립트는 기본적으로 문장들로 이루어져 있으며 문장의 구분은 개행으로 이루어집니다. 개행을 제외한 다른 것으로는 문장을 구분하지 않습니다.

ex)

**동해물과 백두산이  
마르고 닳도록**

위 문장에서 '동해물과 백두산이'와 '마르고 닳도록'은 개행으로 인해 서로 구분되는 문장입니다.

또한 스크립트에는 북마크(책갈피)라는 것이 있습니다. 어떠한 문장의 시작과 끝을 <>로 감싸면 그 문장은 출력되는 문장이 아닌 책갈피로 작동하게 됩니다. 책갈피는 위치를 나타내며 책갈피 바로 아래의 문장과 같은 위치를 가지게 됩니다.

ex)

**하느님이 보우하사  
우리나라 만세  
<애국가 1>  
무궁화 삼천리  
화려강산**

위 문장에서 '애국가 1'이라는 문장은 출력되는 문장이 아닌 '무궁화 삼천리' 문장의 위치를 가르키게 됩니다.

다음에 나오는 것은 시나리오 코드를 작성하는 방법입니다.

시나리오에서 사용 가능한 명령어에는 다음과 같은 것들이 있습니다. 명령들 또한 개행으로 구분되며 한 줄에는 한 명령어와 그 명령어에 사용되는 인수들만 쓸수 있습니다.

Next

이 명령어는 스크립트의 현재 문장을 출력하고 다음 문장으로 이동합니다. 문장의 위치는 시나리오 시작시 첫번째 줄이 디폴트입니다.

## Move

이 명령어는 출력할 문장의 위치를 이동시킵니다. Move 뒤에 띄어쓰기 후 이동할 위치를 지정해주며 위치로 지정해주는 값은 스크립트 상에 존재하는 북마크를 사용합니다.

ex) Move 애국가 1

## StopTime

이 명령어는 게임의 시간을 정지시킵니다. 정지시키는 범위는 {와 }로 지정하며 이 중괄호 역시 한줄을 차지하여야 합니다.

ex)

### StopTime

{        ← 시간이 정지되는 부분

### Next

### Next

}        ← 시간이 다시 흐르는 부분

시간이 정지되면 다음 문장으로 넘어가기 위해 x 키를 사용하여 하며, 시간이 정지되어 있지 않다면 문장들이 시간에 따라 자동으로 넘어갑니다. 디폴트는 시간이 흐르는 상황입니다.

## SetUpper

이 명령어는 문장이 출력되는 대화창을 위로 이동시킵니다. 디폴트는 아래입니다.

## SetDown

이 명령어는 문장이 출력되는 위치를 아래로 이동시킵니다. 디폴트는 아래입니다.

## ChangeLeft 스프라이트

이 명령어는 스프라이트에 입력한 이름의 스프라이트를 어드레서블에서 찾아 왼쪽 초상화로 설정합니다. None 을 입력하면 초상화가 없어집니다. 디폴트는 None 입니다.

## ChangeRight 스프라이트

이 명령어는 스프라이트에 입력한 이름의 스프라이트를 어드레서블에서 찾아 오른쪽 초상화로 설정합니다. None 을 입력하면 초상화가 없어집니다. 디폴트는 None 입니다.

## EnablePortrait

이 명령어는 초상화를 출력합니다. 디폴트는 출력하지 않는 상태입니다.

## DisablePortrait

이 명령어는 초상화를 출력하지 않게 만듭니다. 디폴트는 출력하지 않는 상태입니다.

## HighlightLeft

이 명령어는 왼쪽 초상화를 강조 표시합니다. 초상화는 항상 둘중 하나가 강조상태여야 하며, 디폴트는 왼쪽 초상화 입니다.

## HighlightRight

이 명령어는 오른쪽 초상화를 강조 표시합니다.

## 명령어를 적용하는 법

먼저 규칙에 맞게 시나리오 코드를 작성합니다.

그 다음 유니티에서 메뉴바의 Window/DialogueParser 를 선택해 파서 윈도우를 킵니다. 그 다음 파서 윈도우 창의 코드에 작성한 시나리오 코드를 바인딩 시킵니다. 그리고 아래의 Parse 버튼을 누르면 프로젝트의 Asset 폴더에 시나리오 코드의 이름과 같은 bytes 확장자를 가진 바이트코드가 생성됩니다.

이 바이트코드를 적절한 위치로 이동시킵니다.

바이트코드와 스크립트가 준비되어 있으면 이를 클릭한 후 인스펙터 창의 위에 Addressable 를 체크합니다. 어드레서블에서 관리되는 상태가 된 이 데이터들의 라벨을 바꿔줘야 합니다. 메뉴바의 Window/Asset Management/Addressable/Group 를 클릭해 어드레서블의 그룹관리 창을 킵니다.

그룹관리 창에서 방금 추가한 바이트코드와 스크립트를 찾은 후 뒤의 Label 를 Text 로 바꾸어 줍니다.

마지막으로 대화가 실제로 출력될 유니티의 씬에서 새로운 오브젝트를 만든후 이름을 "DialogueData"로 변경합니다.

그리고 이 오브젝트에 DialogueData 컴포넌트를 추가한 다음 사용할 스크립트와 바이트코드를 인스펙터창에 입력해 줍니다. 이때 입력되는 데이터들은 어드레서블에 추가되어 있어야 합니다.

또한 대화창에서 초상화로 사용될 스프라이트들 또한 어드레서블에 추가되어 있어야 합니다. 이때 스프라이트들은 Label 를 Sprite 로 지정하여야 합니다.