

ToryFramework v1.7.5

익스포트 방법

`ToryFramework` 폴더와 `ToryValue` 폴더를 함께 익스포트하세요.

- 또는 두 폴더를 작업 중인 프로젝트에 복사해도 됩니다.
- `(Exclude This Folder)` 폴더는 의존성이 없으므로 익스포트하지 않아도 됩니다.

주의!!! `ToryFramework` 은 `ToryValue` 에 의존성이 있습니다. 따라서 `ToryFramework` 을 쓰기 위해선 반드시 `ToryValue` 도 함께 익스포트해야 합니다 .

임포트 방법

따로 없습니다. 임포트하면 아래가 자동으로 설정됩니다.

- `ToryFramework` 컴포넌트들의 `Script execution order` 가 `-9998` 로 자동 설정됩니다.
- `Scripting define symbol` 에 `TORY_FRAMEWORK` 심볼이 자동으로 추가됩니다.

주의!!! 업데이트 방법

주의!!! 이전 버전으로부터 판올림할 때, 다음 폴더를 임포트하지 않도록 각별히 주의하세요!

- `Assets/ToryFramework/(Do Not Update This Folder)` 폴더를 임포트하지 마세요! 이 폴더는 에디터를 통해서 프로젝트 별로 커스터마이징한 파일들이 모여있는 곳입니다. 따라서 최초 임포트 시에만 임포트하세요.
- `v1.6.0` 보다 이전 버전으로부터 판올림하는 경우 정정만에게 문의하세요.

프리팸 사용법

`Assets/ToryFramework/Prefabs/ToryFramework` 또는 `Assets/ToryFramework/Prefabs/ToryFramework (with SecureKeysManager)` 프리팸을 씬에 임포트합니다.

- `ToryScene` 은 유니티 씬을 세부적으로 나누어 관리할 수 있는 툴입니다.
 - `TorySceneStateGenerator` 컴포넌트를 통해 커스텀 토리 씬을 만들 수 있습니다.
 - `Animator` 컴포넌트를 통해 토리 씬의 흐름을 제어할 수 있습니다.

주의!!! `TorySceneFlow (Example)` 애니메이터 컨트롤러를 그대로 사용하지 말고, 외부 폴더에 별도로 만들어 사용하세요. 왜냐하면, 업데이트 시에 파일이 덮어쓰워져 문제가 생길 수도 있기 때문입니다.

- 예시를 먼저 보고, 이해가 되지 않는 부분은 정정만에게 물어보세요.
- `ToryInput` 은 다양한 입력을 처리하고, 플레이어와 관련한 이벤트를 제공하는 툴입니다.
 - 다양한 입력 형식을 지원합니다.

- 필터링과 증폭 기능을 제공합니다.
 - 현재 인터랙션이 있는지, 플레이어가 있는지를 판별하고 이벤트를 발생시켜 줍니다.
 - 멀티-인풋을 지원합니다.
- `ToryValue` 는 저장, 불러오기, 초기값과 관련된 기능을 제공하는 툴입니다.
 - `ToryValueGenerator` 를 통해서 커스텀 토리 밸류를 생성할 수 있습니다.
 - 생성된 토리 밸류는 `ToryValueBehaviour` 컴포넌트를 통해서 키, 값, 초기값, 저장값을 설정할 수 있습니다.
 - 저장값은 `PlayerPrefs` 으로 저장되며 항상 싱크됩니다. 저장이 되어 있으면 푸른색, 아니면 붉은색으로 표시됩니다.
 - 저장값은 `PlayerPrefsElite` 를 통해 보안을 보장합니다.
- `SecureKeysManager` 는 `PlayerPrefsElite` 라이브러리를 사용하기 위한 게임 오브젝트입니다. 이 프리팹이 없는 토리밸류가 동작하지 않으므로 씬에 반드시 포함해 주세요.

스크립트 사용법

토리프레임워크의 모든 기능은 스크립트를 통한 접근과 사용이 가능합니다.

접근

- `ToryFramework` 네임스페이스를 사용하면 아래의 싱글톤 인스턴스에 접근할 수 있습니다.
 - `ToryScene.Instance`
 - `ToryInput.Instance`
 - `ToryValue.Instance`
 - `ToryTime.Instance`
 - `ToryProgress.Instance`
- 축약 버전(Shortener)를 마련해 두었습니다. `TF` 키워드를 통해 접근하고, 아래 예시처럼 사용합니다.
 - `TF.Scene` (= `ToryFramework.ToryScene.Instance`)

이벤트 사용

- `OnEnable()` 에서 이벤트를 등록하고, `OnDisable()` 에서 해제하여 사용합니다.

```

1  void OnEnable()
2  {
3      TF.Scene.Title.Updated += UpdateTitle;
4  }
5  void OnDisable()
6  {
7      TF.Scene.Title.Updated -= UpdateTitle;
8  }
9  void UpdateTitle()
10 {
11     // The code here will be called every frame only in the title
    scene.
12 }
```