

UI 최적화 방향성

History

업데이트 일자	작성자	내용
2017/08/01	박정민	기본 정리.

1. Panel 정리 및 Depth 관리.

- 기존에 Depth관리로 처리할 수 있는 많은 부분들이 Panel로 구현되어 있음.
- Scroll View, Popup을 제외한 대부분의 것들을 하나의 Panel안에서 처리 가능.
- 루트(현 GameManager안의 UI3dPivot, UIPivot, EternalPivot)에 패널을 배치하고 추가 패널이 필요한 부분들만 Panel 할당.
- 각 아틀라스, 폰트별 영역대 설정 후 한 패널 안에서 해당 영역대에 배치.
- [UI 최적화 가이드_Panel_AtlasDepth 영역대 설정] 문서 참조. (아트도 공유)

2. Pooling, Instantiate 사용

- Asset.Load, Resources.Load를 최소화하고, Instantiate 사용.
- Binder, Slot, ListItem 등으로 불러올 때, Object Pool 이용
- Object Pool에서 최초 1회 불러올 때만 load 사용, 그 이후는 Instantiate

3. UIBinder : 최소한으로 사용, UIListItem 정리

- 빈번한 GetComponent<T> 사용 지양.

4. 카메라 관리들 필요.

- 카메라 Component에 Culling Mask 명확하게 설정.
- UICamera Component에 Event Mask 및 Event Source 정리.
- 필요한 카메라만 On/Off하는 관리 툴 필요.
- UI에서 가급적 뒷배경이 안나올 수 있게 처리 필요. (InGame Camera와 병행해야 하는 경우를 제외하고 뒷배경을 안쓰는 것을 지향.)
- UI 뒷배경에 Alpha처리 되어있는 부분 처리 필요.

5. NGUI Event type 3D -> 2D : 추후 작업

- 미세한 퍼포먼스 상승.
- UI 내에 3D 오브젝트 입력을 받아와야 하는 경우가 현재까지 없음.
- 추후 3D 오브젝트 입력을 받아와야 하는 경우가 생기면, 해당 부분에 맞게 카메라를 추가하는 방식으로 처리

6. Event Custom Script : 추후 작업

- 미세한 퍼포먼스 상승.
- UIButton, UIEventTrigger 등 필요한 이벤트에 비해 너무 많은 이벤트 발생.
- Custom Script로 필요한 이벤트만 받아서 처리.

7. 스크립트 내부에 동적 할당하는 "string" 제거.

- 상시 작업