

6주차 정리

개요

UI/UX 디자인 기초 학습

유니티 내 UI 요소 생성과 상호작용 실습

사용자 인터페이스 디자인 및 플로우 구현

개요

- ▶ UI/UX 디자인 기초 학습
- ▶ 유니티 내 UI 요소 생성과 상호작용 실습
- ▶ 사용자 인터페이스 디자인 및 플로우 구현

▼ UI/UX 디자인 기초 학습

- UI와 UX의 차이:
 - UI (User Interface): 사용자 인터페이스로, 앱 또는 웹사이트와 같은 디지털 제품의 시각적인 요소를 의미합니다. 버튼, 아이콘, 색상, 레이아웃 등이 모두 UI에 해당됩니다.
 - UX (User Experience): 사용자 경험으로, 디지털 제품을 사용하는 사용자들이 느끼는 전반적인 경험을 의미합니다. 사용의 편의성, 효율성, 만족도 등이 모두 UX에 영향을 미칩니다.
- 디자인 원칙:
 - 가독성: 사용자가 정보를 쉽게 읽고 이해할 수 있도록 폰트, 텍스트 크기, 줄 간격 등을 신경써야 합니다.
 - 간결성: 필요한 정보와 기능만 제공하여 복잡성을 줄입니다.
 - 일관성: UI 요소들 사이의 일관된 디자인과 동작을 유지하여 사용자의 혼란을 방지합니다.
 - 표준화: 흔히 사용되는 디자인 패턴과 관습을 따르는 것이 사용자들에게 익숙함을 제공합니다.
- 컬러와 타이포그래피:

- 컬러 선택: 사용자에게 원하는 느낌과 어울리는 적절한 컬러를 선택하는 것이 중요합니다. 각 색상이 어떤 감정과 연관되는지 이해하는 것이 도움이 됩니다.
- 타이포그래피: 적절한 폰트 선택과 텍스트의 크기, 라인 간격 등을 고려하여 가독성과 디자인의 일관성을 유지합니다.
- 레이아웃과 그리드 시스템:
 - 레이아웃: UI 요소들을 배치하는 방법으로, 적절한 공백과 요소들 간의 관계를 고려하여 균형 잡힌 레이아웃을 구성합니다.
 - 그리드 시스템: UI 요소들을 배치하는 데 사용되는 구조로, 일관된 간격과 비율을 유지하여 일관성과 조화를 제공합니다.
- 사용자 중심 디자인:
 - 사용자 조사: 사용자의 요구사항과 선호도를 이해하기 위해 조사를 진행합니다.
 - 프로토타이핑: 디자인 아이디어를 빠르게 시각화하여 사용자와 테스트하고 피드백을 수집합니다.
 - 반복적 디자인: 사용자 피드백을 수용하여 개선하는 과정을 반복하여 최적의 디자인을 찾아냅니다.

▼ 유니티 내 UI 요소 생성과 상호작용 실습

단계 1: 새로운 UI 요소 생성하기

1. Unity를 실행하고, 새로운 2D 또는 3D 프로젝트를 생성합니다.
2. 프로젝트 화면에서 "GameObject" 메뉴를 클릭하고, "UI"를 선택합니다.
3. 원하는 UI 요소를 선택하여 생성합니다. 주로 사용하는 UI 요소에는 다음과 같은 것들이 있습니다:
 - Canvas: UI 요소들이 들어갈 컨테이너 역할을 합니다. 기본적으로 생성되는 UI 요소들은 모두 Canvas 내부에 생성되어야 합니다.
 - Panel: 배경이나 레이아웃을 구성할 때 사용됩니다.
 - Text: 텍스트를 표시하기 위해 사용됩니다.
 - Image: 이미지를 표시하거나 버튼을 만들 때 사용됩니다.
 - Button: 클릭 가능한 버튼을 생성합니다.

단계 2: UI 요소 배치 및 설정

1. Canvas를 선택하고, 씬 뷰에서 적절한 위치로 이동시킵니다.

2. 다른 UI 요소들도 마찬가지로 씬 뷰에서 적절한 위치로 이동시킵니다.
3. 선택한 UI 요소의 Inspector 창을 열고, 속성을 설정합니다. 예를 들어, Text 요소라면 텍스트 내용, 폰트 크기, 색상 등을 설정할 수 있습니다.

단계 3: UI 요소 간 상호작용 구현

1. Button을 생성하고 버튼의 텍스트를 설정합니다.
2. 해당 버튼의 스크립트를 작성하기 위해 C# 스크립트를 생성합니다. Project 창에서 우클릭하고, "Create" > "C# Script"를 선택합니다. 이름을 지정한 후에 열어서 코드를 작성합니다.

```
using UnityEngine;
using UnityEngine.UI;

public class ButtonClickHandler : MonoBehaviour
{
    // 버튼 클릭 시 실행될 메소드
    public void OnButtonClick()
    {
        Debug.Log("버튼이 클릭되었습니다!");
    }
}
```

1. 해당 버튼에 방금 만든 스크립트를 추가합니다. 버튼을 선택한 상태에서 Inspector 창에 있는 "Add Component" 버튼을 누르고, 방금 작성한 스크립트를 검색하여 추가합니다.
2. 버튼을 클릭했을 때 원하는 동작을 스크립트 내부에 구현합니다. 위의 예시 코드에서는 버튼이 클릭될 때 "버튼이 클릭되었습니다!"라는 로그를 출력합니다.

이제 버튼을 실행하여 클릭하면 로그가 출력됩니다. 이와 같은 방식으로 다양한 UI 요소를 생성하고 상호작용을 구현할 수 있습니다. 더 복잡한 상호작용을 위해서는 Unity의 UI 이벤트 시스템에 대해 더 자세히 학습하는 것이 도움이 될 수 있습니다. UI 이벤트 시스템을 활용하면 버튼 클릭, 드래그, 터치 등 다양한 사용자 입력에 대응하는 상호작용을 만들 수 있습니다.

▼ 사용자 인터페이스 디자인 및 플로우 구현

워크플로우 구현

