[2023-09-06-수요일]

[한 일]

로켓 리그 GitHub 생성 & 브랜치 연결

로켓 리그 UI 이미지 파일 작업

로켓 리그 타이틀 씬 작업

DOTween 에셋 추가

[타이틀 씬 로직]

title\_bg-01.png 출력

비동기 로딩

로딩 완료시 4초 안에 끝나도록 설정

아닐 경우 대기

비동기 로딩 완료 후

1초간 페이드 아웃

로고 출력

1초간 페이드 인

1초 대기

3초간 스케일 105% 변경

1초간 페이드 아웃

[로비씬 로직]

동시에 동작

검은 배경 레이어 출력(시작 알파 100%)

0.5초간 페이드 아웃

시작 문구 출력

0.5초간 Pos Y 아래에서 위로 이동

While로 색 변화 효과 출력

로켓리그 로고 출력(시작 알파 0%)

0.5초간 페이드 인 & 위에서 아래로 이동

터치할 경우

UI 삭제

[문제점]

1. UI 이미지를 이미지 컴포넌트에서 가져올 수 없는 문제가 발생함.
2. TMP Text의 Glow 색상을 DOTween을 사용하여 변경하려 했으나 일반적인 방법으로 변경할 수 없음.

[해결]

1. 해당 이미지의 인스펙터 창에서 Texture Type을 Sprite (2D and UI)로 수정하니 정상적으로 동작한다.
2. TMP\_Text의 Material을 TMP\_Text.fontSharedMaterial로 가져온 후 가져온 Material의 shader를 Material.shader로 가져온다. 그리고 Shader.PropertyToId(“\_GlowColor”)로 GlowColor의 아이디를 받아온 후 DOTween.To()라는 DOTween용 람다 함수로 아래와 같이 색상을 변경하니 됐다. DOTween.To(( => material.GetColor(GlowColorID), color => material.SetColor(GlowColorID, color), targetColor, t); 로 지정한 시간동안 색상을 변경하게 한다.

[팁]

1. 0과 1사이의 랜덤한 소수를 얻고자 할 때 Random.value를 사용하면 된다.

################################################

[2023-09-07-목요일]

[한 일]

로비 게임 로고 효과 작업

로비 GUI 작업

로비 버튼 효과 작업

IPointerEnterHandler, IPointerExitHandler 인터페이스를 사용하여 마우스 포인터가 Enter, Exit 할 때 버튼과 텍스트의 색상을 바꾸게했다.

커스터마이징에 대해 의논함

Resources 폴더에 Prefab으로 자동차의 파츠를 등록 한 후 해당 사용자의 커스터마이징에 따라 자동차를 생성 하는 로직

ex) 스크립트로 빈 오브젝트를 생성한 후 사용자의 요구에 따라 하나씩 Prefab을 불러와서 자식으로 추가한다.

[문제점]

A 씬에서 만든 캔버스를 B 씬으로 복사 했을 때 버튼이 제대로 동작하지 않는 현상

[해결]

캔버스를 붙여넣기 할 씬에 빈 캔버스를 생성한 후 삭제한다. 그러면 정상적으로 버튼이 동작한다.

[팁]

Source Tree에서 병합(Merge)을 하고 Pull을 받을 때 병합한 브랜치로 설정해서 내려 받아야한다.

[2023-09-08-금요일]

[한 일]

프로젝트 병합

자동차가 벽에 닿을 경우 벽을 탈수 있게 기능 구현시도

팀 기능 구현 과제 선정

유순 : 공 스킬

경민 : 차 공중 관련 기능

준호 : 차 스킬

나 : 차 커스터마이징 기능

[문제점]

자동차가 벽에 닿을 경우 transform 값이 이상해진 상태로 벽을 타게 된다.

[해결]

벽에 닿을 경우 중력 방향을 다르게 설정. 하지만 좌/우 회전이 되지 않는 현상이 발생함. 나중에 폴리싱 기간에 작업하기로 일정 변경

[팁]

중력을 원하는 방향으로 바꿀 수 있다.

[2023-09-11-월요일]

[한 일]

CSV 파일 생성

커스터마이징 매니저 제작 시작

[커스터마이징 매니저 로직]

자동차의 기본 구조는 아래와 같다.

Player

Collider

Kart

CarFrames

카 프레임들…

Wheels

Wheels\_FL

Wheels\_FR

Wheels\_BL

Wheels\_BR

Marks

Flags

Colors

Camera

위의 구조에서 복수 표현이 사용되는 오브젝트 하위에 모든 차량 파츠 프리팹을 추가하고 상황에 따라 Active를 True/False해서 오브젝트 풀링한다.

Photon PUN 서버와 연결될 때를 잘 생각해야 한다.

\*Active 변동 여부는 자동으로 동기화 되지 않으므로 따로 RPC로 정보를 보내줘야 한다.

[문제점]

CSVReader 스크립트에서 CSV파일을 불러올 때 키가 중복된다는 오류 발생

Photon PUN 서버와 연결했을 때 오브젝트 Active 값을 제대로 전달 할 수 없는 문제가 발생할 예정.

[해결]

사용자의 오브젝트 풀링 정보를 저장하고 원하는 타이밍에 RPC 할 수 있는 함수를 만들 예정.

[팁]

[2023-09-12-화요일]

[한 일]

[문제점]

1. CSV 파일을 읽어온 후 딕셔너리에 저장할 때 중복된 키 값이라고 오류가 발생한다.하지만 로직에는 문제가 없다.
2. CSV 파일들을 기반으로 생성한 dataDictionary의 키 값이 일치함에도 접근할 수 없는 오류가 발생

2-1. 하지만 그래도 접근할 수 없는 문제가 발생

1. CSV 파일을 읽어올 때 공백이 생기는 현상이 발생하여 Prefab을 Resource 폴더에서 검색하고 생성할 때 문제가 생김
2. 오브젝트 토글을 위해 플레이어가 가지고 있는 오브젝트 중에 원하는 오브젝트를 호출 할 때 플레이어 오브젝트가 계층 구조로 부모/자식이 나뉘어 있어 직계 자식만 가져올 수 있는 Transform.Find()함수를 사용할 수 없어 원하는 오브젝트를 호출할 수 없는 문제가 발생

[해결]

# 1. ColorList라는 CSV 파일이 손상되어 있어 새로운 파일로 대체하니 정상적으로 작동한다.

2. CSV Reader에서 ‘\n’으로 행을 구분하지 않아 행과 열의 값이 합쳐져서 키 값에 오류가 발생해 생긴 문제였다. 그래서 ‘\n’으로 행을 구분했다.

2-1. 문제를 해결하기 위해 노력한 결과 데이터 구조가 복잡해서 일반적인 Dictionary접근법으로는 접근할 수 없어 Foreach문을 사용하여 Key값을 통해 접근하니 잘된다.

3. Trim()함수를 사용하여 강제로 공백을 제거하니 해결됨.

4. 원하는 오브젝트를 찾을 때 까지 모든 계층의 자식 오브젝트들을 검색하는 아래와 같은 재귀 함수를 사용하여 문제 해결.

Transform FindChildRescursive(Transform parent, string targetName)

Foreach (Transform child in parent)

// 원하는 자식 오브젝트를 찾은 경우

if (child.name == targetName)

{

// 찾은 자식 오브젝트 반환

return child;

}

// 찾지 못한 경우

else

{

// 모든 하위 자식 오브젝트를 검색 하기 위해 재귀 호출

// FindChildRescursive()의 매개변수로 child를 넣어 계층 구조를 탐색하게 한다.

Transform foundChild = 현재함수(child, targetName)

If (foundChild != null)

{

// 찾은 자식 오브젝트 반환

return foundChild;

}

}

}

// targetName과 일치하는 자식 오브젝트를 찾지 못한 경우

// 재귀를 위해 null 반환

return null;

}

[팁]

계층 구조의 오브젝트를 찾을 때 재귀 함수가 유용하다.

[2023-09-13-수요일]

[할 일]

커스터마이징 정보를 저장하는 PlayerPrefab 스크립트 만들기

커스터마이징 UI 구현

Photon PUN 서버와 연동되도록 코드 수정

[한 일]

1. PlayerDataManager 스크립트 만들기

[목적]

: CustomizingManager 클래스에 있는 정보를 PlayerDataManager에 저장하고 사용자가 요청할 때 PlayerPref에 저장되어 있는 정보를 자신과 다른 사용자에게 RPC로 전달해서 오브젝트를 구현하는 목적.

[방법]

**PUN 네트워크 플로우**: PUN을 사용하면 게임 플레이어 간에 네트워크 메시지 및 RPC를 통해 통신하고, 이를 통해 오브젝트 생성 및 동기화를 한다. 이러한 네트워크 플로우를 활용하여 플레이어 커스터마이징 정보를 다른 플레이어에게 전달하고 원하는 씬에서 오브젝트를 생성하는 방법.

[오브젝트 생성법]

커스터마이징에서는 오브젝트 풀링을 사용해 모든 오브젝트를 active false로 추가하지만 커스터마이징 적용을 하면 적용된 파츠만 자동차에 추가한다. 그리고 자동차 오브젝트를 생성할 때 CraFrames, Wheels의 자식이 없는 상태로 생성한 후 파츠를 추가한다.

[달성]

1. CustomizingManager에 플레이어의 현재 모든 파츠의 인덱스를 저장 한 후 PlayerDataManager에서 PlayerPref에 저장/호출 기능 완료

[\*참고]

디버그를 위해 오브젝트 최초 생성시 active true로 임시 설정

2.CustomizingManager에서 PlayerPrefab에 저장된 정보를 바탕으로 자동차 오브젝트를 생성하는 함수 만들 예정

[문제점]

[해결]

[팁]

DontDestroyOnLoad() 함수를 사용하면 다른 씬으로 이동해도 오브젝트가 파괴되지 않게 할 수 있다.

PlayerPref에서 값을 호출할 때 PlayerPref.GetInt(Key, -1)과 같이 오버로드를 하면 값 호출에 실패했을 때 -1를 반환한다.

[2023-09-14-목요일]

[할 일]

커스터마이징 씬 구현하기

Photon PUN 서버에서 잘 적용되는지 확인하기

[한 일]

[문제점]

[해결]

[팁]

ExitGames.Client.Photon.Hashtable을 사용하면 같은 포톤 서버에 접속해 있는 모든 유저들 끼리 정보를 공유할 수 있다. PlayerPref의 멀티판이라고 보면 된다.

사용법은 ExitGames.Client.Photon.Hashtable customProperties(객체명) = new ExitGames.Client.Photon.Hashtable (); 로 객체를 생성한 후에 해당 객체에 customProperties.Add(“Key”,value);를 사용하여 값을 넣을 수 있다. 데이터 타입은 자유다. 그리고 호출 할 때는 (데이터타입) customProperties(객체명)[“Key’];로 가져온다. 인터페이스로 OnPlayerPropertiesUpdate()함수를 오버라이드 해서 사용하면 커스텀 프로퍼티의 값이 바뀔 떄 마다 해당 함수가 콜백되는데 이 때 함수의 매개변수인 ExitGames.Client.Photon.Hashtable changedProps 즉 changedProps를 사용해서 (int)changedProps[“키값”]과 같은 방법으로 정보를 가져올 수 있다. 이 떄 매개변수로 받아오는 changedProps는 업데이트된 해쉬테이블이다.