

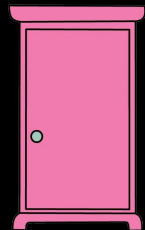
PORTAL 2™



PORTAL 2

TEAM : 어디로든 문

정하영, 김나영, 배주백



1. 게임 소개

게임의 특징

- 대규모 싱글 플레이어: 다음 세대의 게임 플레이와 마음을 단숨에 휘어잡는 스토리를 제공합니다.
- 완벽한 2인 협동 게임: 멀티 플레이어 게임은 그만의 고유한 스토리와 캐릭터, 게임 플레이를 제공합니다.
- 더욱 발전한 물리 엔진: 더 거대한 범위의 흥미로운 도전을 창조할 수 있게 하지만, 게임 자체에 부담이 가지는 않습니다.
- 독창적인 음악
- 대서사시의 속편: 오리지널 포털은 전세계 30개 이상의 매체가 선정한 '2007년 올해의 게임'의 반열에 올랐습니다.
- 편집 도구: 포털 2 편집 도구가 포함될 예정입니다.

- 스팀 포털 2 상점 페이지의 '게임에 대해' 설명문

2. 역할

정하영

플레이어 구현
포탈 시스템 구현
게임 시작 및 종료 설계



김나영

레이저 구현
레이저와 상호작용구현
맵디자인 연구
메테리얼 최적화 연구



배주백

라이브러리 구현
스위치 구현
점프대 구현
맵 설계



갭트 차트

[illegible]

2. 프로토 문서

버그 트래킹

구분	이슈	진행상황	발생 버전	재현율	작성자	담당자	수정 날짜	수정 확인 날짜
1	그랩한 상태의 큐브를 포탈에 비비적 거리면 그랩을 놓았을때, 사라진다	대기 ▼	프로토	100%	배주백	정하영		
2	GRAP 한 CUBE가 벽을 통과하는 이슈	대기 ▼	프로토	100%	배주백	정하영		
3	플레이어를 포탈에 비추었을 때, 팔만 보인다,	완료 ▼	프로토	100%	배주백	정하영	08/16	08/16
4	포탈에서 반사되는 포탈이 안 보인다	대기 ▼	프로토	100%	배주백	정하영		
5	시야를 위를 보면 총이 사라진다. (총이 화면에서 일정 부분 유지가 되지 않음)	완료 ▼	프로토	100%	정하영	정하영	08/16	08/16
6	큐브가 있는 바닥에 포탈을 쓰면 큐브가 출구로 떨어지지 않음.	완료 ▼	프로토	100%	정하영	정하영	08/16	08/16
7	바닥에 포탈 쓰면 충돌 한번 일어날 경우 큐브가 복제되서 두 개가 공중에 떠다님	대기 ▼	프로토	100%	정하영	정하영		
8	PlayerStart 위치가 바뀔 경우 팔 꺾임	대기 ▼	프로토	100%	정하영	정하영		
9	큐브를 들고 포탈 통과할 경우 큐브 그랩 위치가 바뀜	대기 ▼	프로토	100%	정하영	정하영		
10	큐브가 점프대를 통과할 경우 포탈 반대로 떨어짐	대기 ▼	프로토	100%	정하영	정하영		
11	캐릭터가 포탈 통과할 때, 통과는 되는데 뭔가 위치가 잘못떨어질때도 있고 impulse가 없다	대기 ▼	프로토	100%	정하영	정하영		
12	레벨에 포탈이 배치되어 있지 않으면 포탈건 작동 안함	대기 ▼	프로토	100%	정하영	정하영		
13	기울어진 벽에 있는 포탈로 들어가면 플레이어도 기울어져서 나옴	완료 ▼	프로토	100%	정하영	정하영	08/17	08/17
14	플레이어 포탈 이동 시 운동량 보존 안됨	대기 ▼	프로토	100%	정하영	정하영		
15	큐브 포탈 이동 시 운동량 보존 안됨	완료 ▼	프로토	100%	정하영	정하영	08/17	08/17
16	큐브 포탈 이동 시 방향 반대로 나옴	완료 ▼	프로토	100%	정하영	정하영	08/17	08/17
17	포탈 쓸 수 있는 벽이 plane이 아닐 경우 반대쪽 포탈이 비치지 않음	완료 ▼	프로토	100%	정하영	정하영	08/17	08/17
18	캐릭터가 용암에 닿아도 안 죽음, Tag가 CapsuleCollider에 있는데, Actor에 Player 태그가 있어야 한다	완료 ▼	프로토	100%	배주백	정하영	08/16	08/16
19	큐브를 그랩상태에서 점프대를 타서 큐브가 점프대에 먼저 닿으면 플레이어를 빼고 큐브만 점프대로 이동한다	진행중 ▼	프로토	100%	배주백	배주백		
20	큐브를 그랩상태에서 점프대에 닿게해서 큐브가 점프를 하면 더이상 포탈건 발사 불가	완료 ▼	프로토	100%	배주백	배주백	08/16	08/16
21	점프대를 통과한 큐브가 벽을 투과한다	완료 ▼	프로토	100%	배주백	배주백	08/16	08/16
22	C++을 상속한 블루프린트파일이 언리얼 프로젝트를 재시작하면 InvalidClass 상태가 된다	완료 ▼	프로토	100%	배주백	배주백	08/12	08/13

3. 정하영 담당 구현

프로젝트 담당 파트

간트 차트



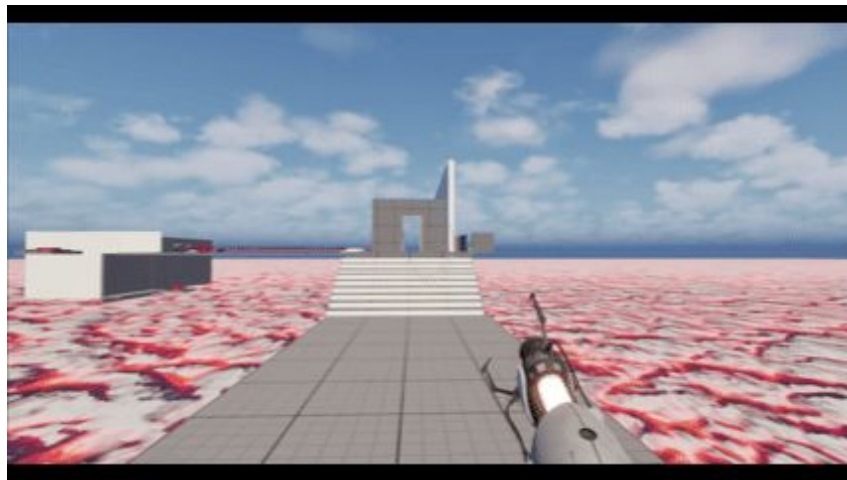
						4	5	6	7	8	9	10	11	12	13	14	15	16	17	
No	대분류	소분류	담당자	우선순위	진행도	기획	프로토													
						월	화	수	목	금	토	일	월	화	수	목	금	토	일	
1	구현 기술RND	포탈건 토이 프로젝트A 개발	정하영	상	완료							일								
4	프로젝트 개설	프로젝트 개설 및 공유	정하영	상	완료								월							
10	플레이어 컨트롤	포탈건 제작	정하영	상	완료															
11	플레이어 컨트롤	플레이어 애니메이션 구현	정하영	상	완료															
12	플레이어 컨트롤	플레이어 조작	정하영	상	완료															
13	플레이어 컨트롤	포탈 구현	정하영	상	완료															
14	플레이어 컨트롤	큐브 구현	정하영	상	완료															
15	포탈 상호작용	포탈과 사람의 상호작용	정하영	상	완료															
16	포탈 상호작용	포탈과 큐브의 상호 작용	정하영	상	완료															
17	포탈 상호작용	포탈과 물체의 상호 작용	정하영	상	알파 진행															
19	포탈 상호작용	포탈 생성 불가 벽	정하영	상	완료															
43	UI	게임 시작-종료 UI 구성	정하영	중	완료															
44	UI	에임 UI	정하영	중	완료															
46	기능 개선	큐브 오브젝트 벽 버그 확인	정하영	하	알파 진행															
47	기능 개선	큐브가 있는 바닥에 포탈 생성시 상호작용	정하영	하	완료															
48	기능 개선	물건을 든 상태에서 포탈건 발사 불가	정하영	하	완료															
49	프로토 발표	프로토 발표 문서 작성A	정하영	상	완료															

핵심 기능 구현 내용

플레이어 구현
포탈 시스템 구현
포탈건 구현
시작 및 종료 UI 설계

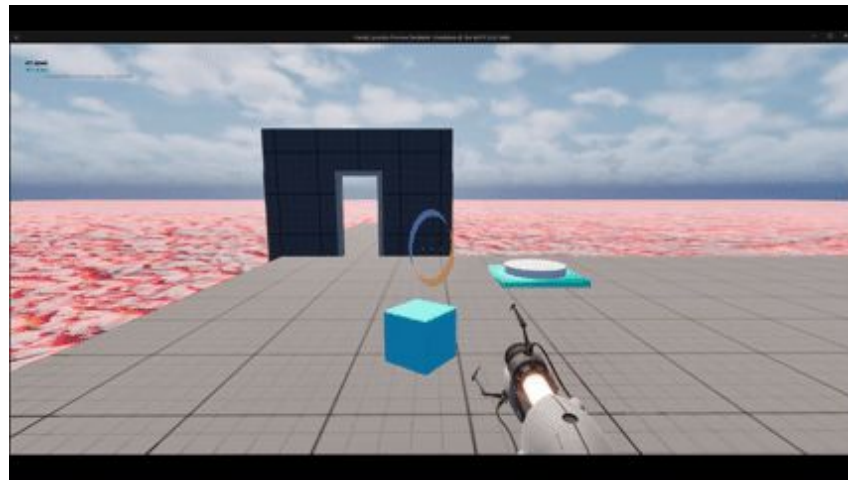
4. 플레이어 구현

1인칭 시점 설정



4. 플레이어 구현

물체 상호작용



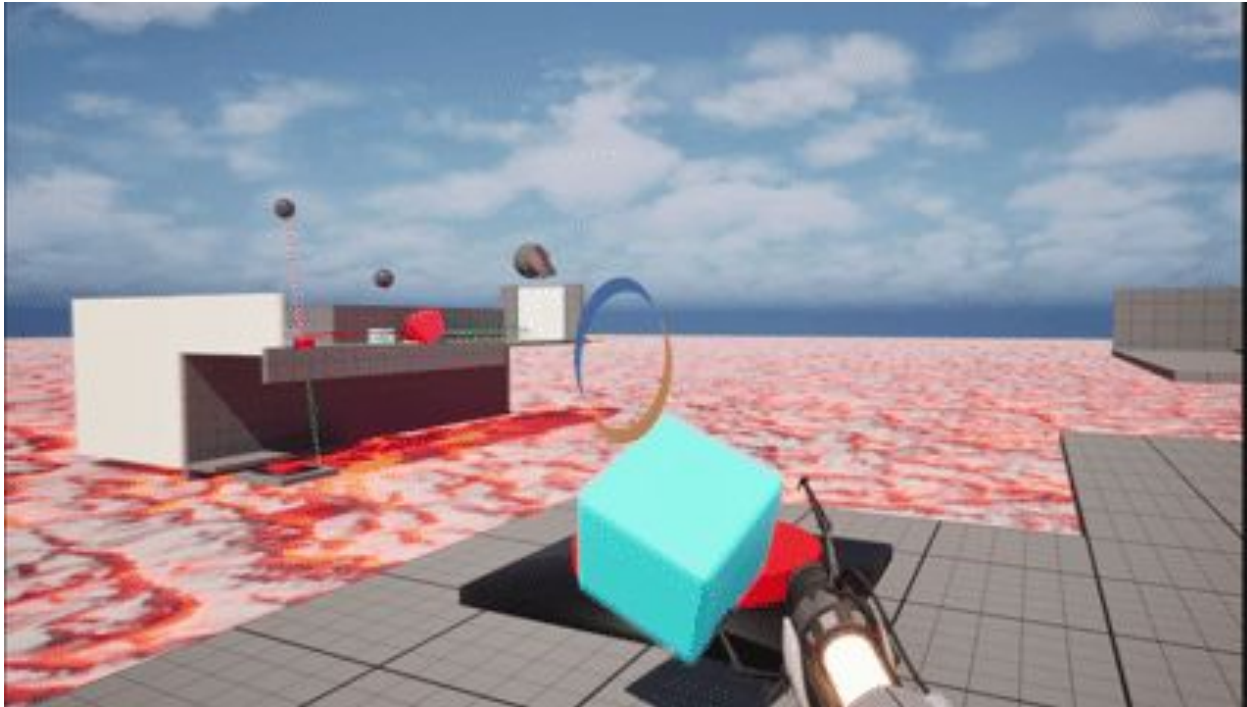
4. 플레이어 구현

포탈건



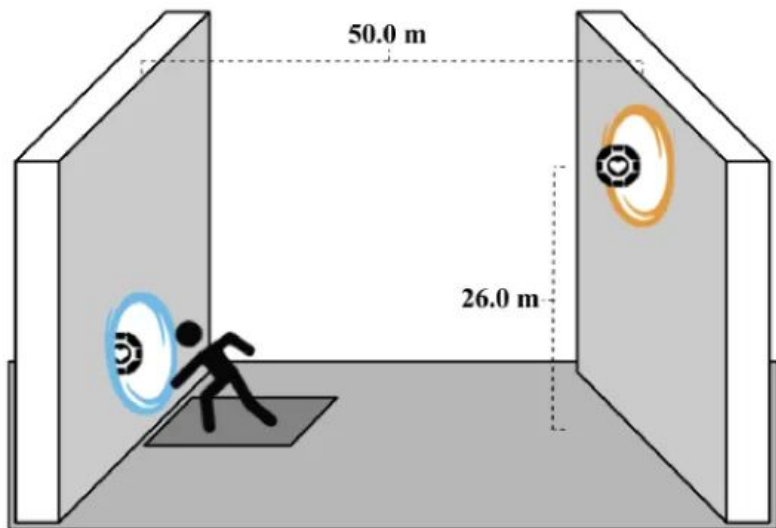
5. 포탈 시스템

포탈 이동 시 운동량이 보존되어야 한다



5. 포탈 시스템

포탈 이동 시 운동량이 보존되어야 한다



// 포탈을 통한 이동 시, 속도(운동량)은
보존되지만 방향은 바뀌어야 한다!

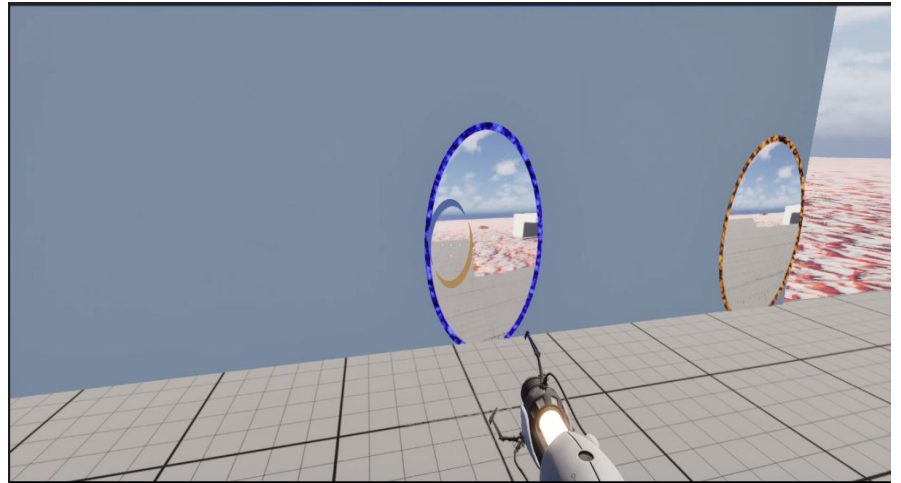
포탈 입구에서 속도의 크기(스칼라값)를 저장

속도의 방향을 입구 포탈 → 출구 포탈 좌표계로
변환

출구의 속도 = 출구 방향 단위벡터 * 입구 속도의
크기

5. 포탈 시스템

반대 포탈의 전방 장면을 보여주어야 한다



5. 포탈 시스템

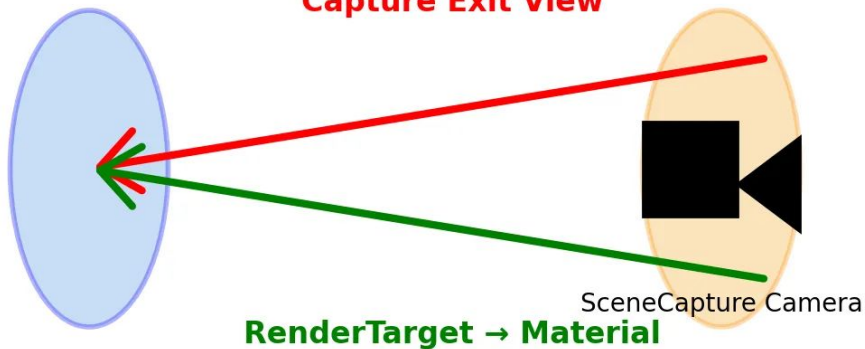
반대 포탈의 전방 장면을 보여주어야 한다

Portal Camera Capture System

Entrance Portal

Exit Portal

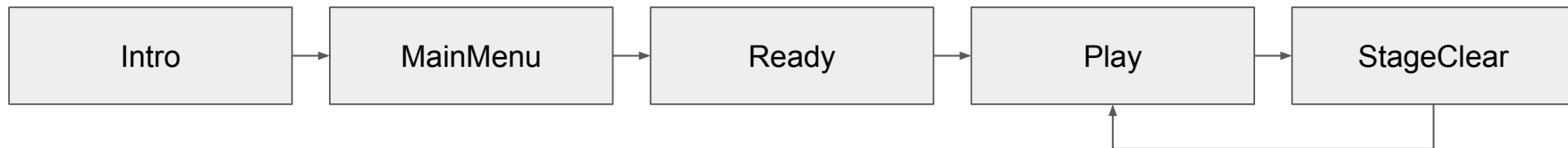
Capture Exit View



출구 포탈 위치에 카메라를 배치

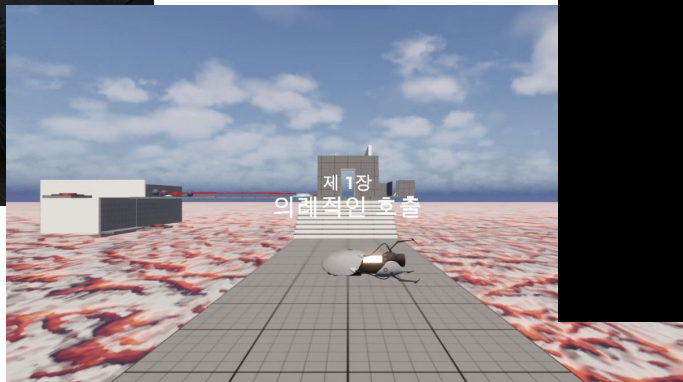
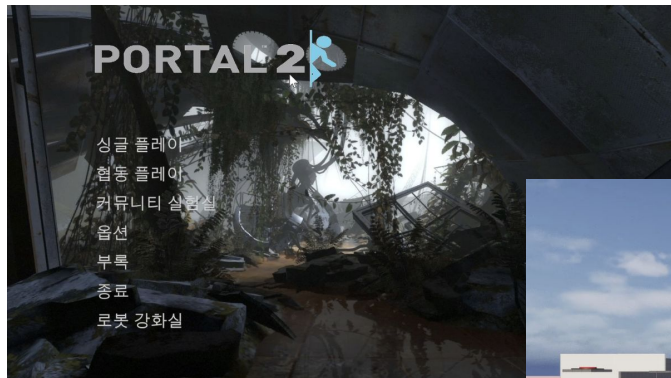
카메라가 찍은 장면을 입구 포탈 표면에
머티리얼로 투사

6. 게임 시작 종료 구성



게임 시작 버튼 클릭 시

다음 스테이지로 이동
(테이블로 관리할 예정)

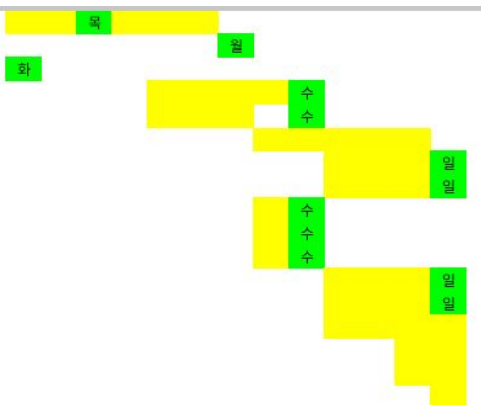


7. 김나영 담당 구현

프로젝트 담당 파트

간트 차트

2	구현 기술RND	포탈건 토이 프로젝트B 개발	김나영	상	완료
5	프로젝트 개설	프로젝트 개설 및 공유	김나영	상	완료
7	리소스 제작	포탈건 제작	김나영	상	완료
8	레이저 기술	레이저 구현 연구	김나영	상	완료
9	레이저 기술	레이저 생성	김나영	상	완료
18	포탈 상호작용	포탈과 레이저의 상호 작용	김나영	상	일파 진행
20	레이저 상호작용	레이저와 Door의 연동	김나영	상	완료
21	레이저 상호작용	레이저와 스위치 상호작용	김나영	상	완료
22	레이저 상호작용	레이저와 오브젝트상호작용	김나영	상	완료
23	레이저 상호작용	레이저 스폰너와 스폰사이의 상호작용	김나영	상	완료
24	레이저 상호작용	레이저와 큐브 상호작용	김나영	상	완료
25	레이저 상호작용	레이저와 타겟 상호작용	김나영	상	완료
26	레이저 상호작용	레이저와 반사벽 상호작용	김나영	상	완료
27	맵 상호작용	터렛 레이저 부착	김나영	중	일파 진행
43	맵 제작	맵 디자인 연구	김나영	중	진행중
44	맵 제작	메테리얼 최적화 연구	김나영	중	진행중
52	프로토 발표	프로토 발표 문서 작성B	김나영	상	진행중



핵심 기능 구현 내용

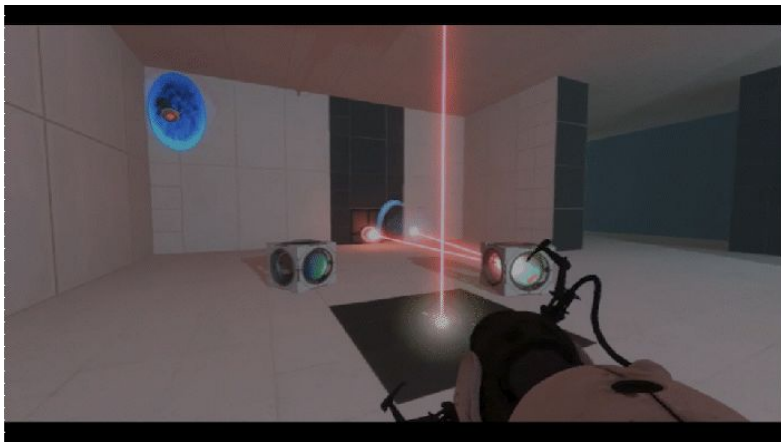
레이저구현
레이저와 오브젝트간의 상호작용
맵 디자인 연구
Material 최적화 연구

8. 레이저 구현

LineTrace 를 활용한 laser 구현방식

$$P(t) = O + t * D$$

원본(원)

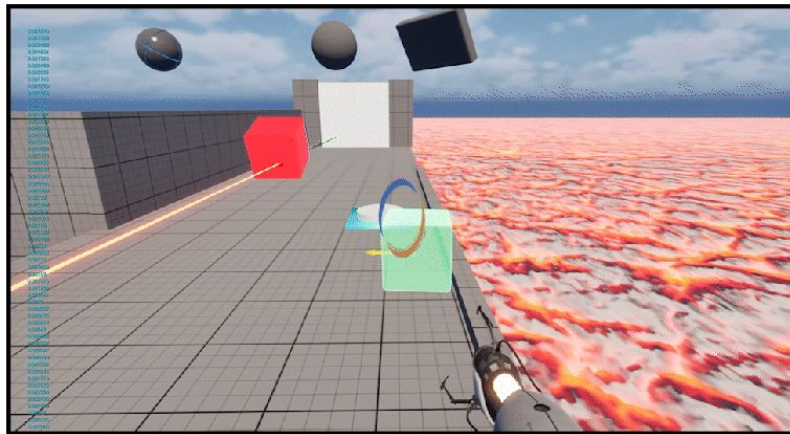


광선 벡터를 나타낸 수식

- O = origin (시작점)
- D = direction(방향)
- t = 스칼라 값(0이면 시작점)

***어느 방향으로 무한히 쏘는
직선(광선)을 나타낸다.**

구현물



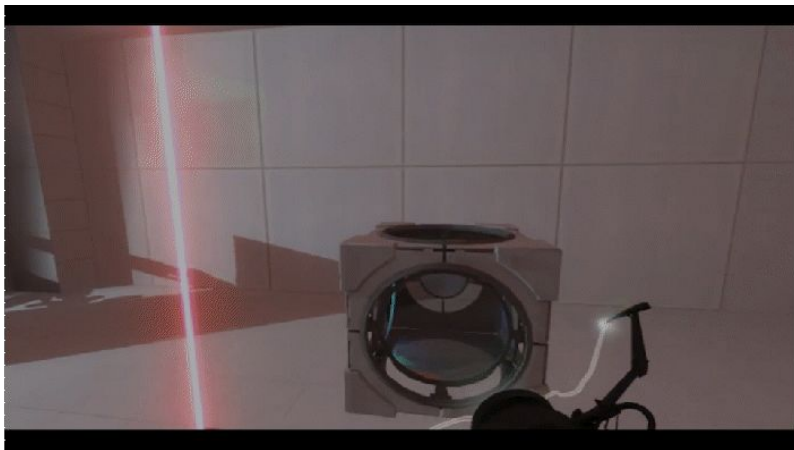
9-1. 레이저 상호작용

레이저와 큐브사이의 상호작용

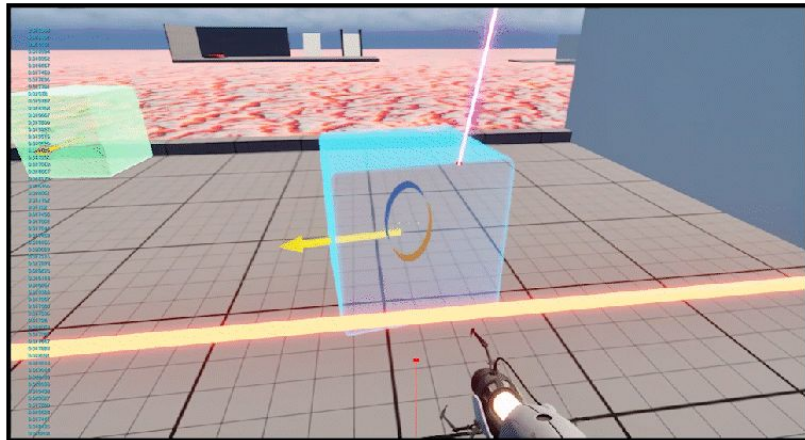
방향벡터를 고정시켜, 그 방향으로만 변화없이 직진한다.

반복적인 반사를 통해 다양한 퍼즐을 연출 할 수있다!

원본(왼)



구현물



9-2. 레이저 상호작용

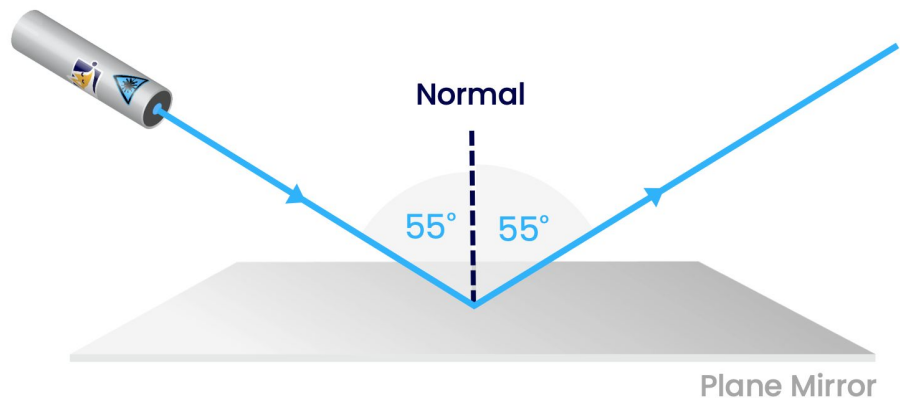
레이저와 반사벽 사이의 상호작용

반사벡터의 이용!(거울효과)

$$R = D - 2(D \cdot N)N$$

- R = 반사벡터(결과방향)
- D = 입사벡터(광선이 들어오는방향)
- N = 법선벡터(벽의 표면이 수직으로 향하는방향)
- $D \cdot N$ = 내적 (수직 성분크기)

"수직 성분만 반대로 뒤집는 과정"



구현물

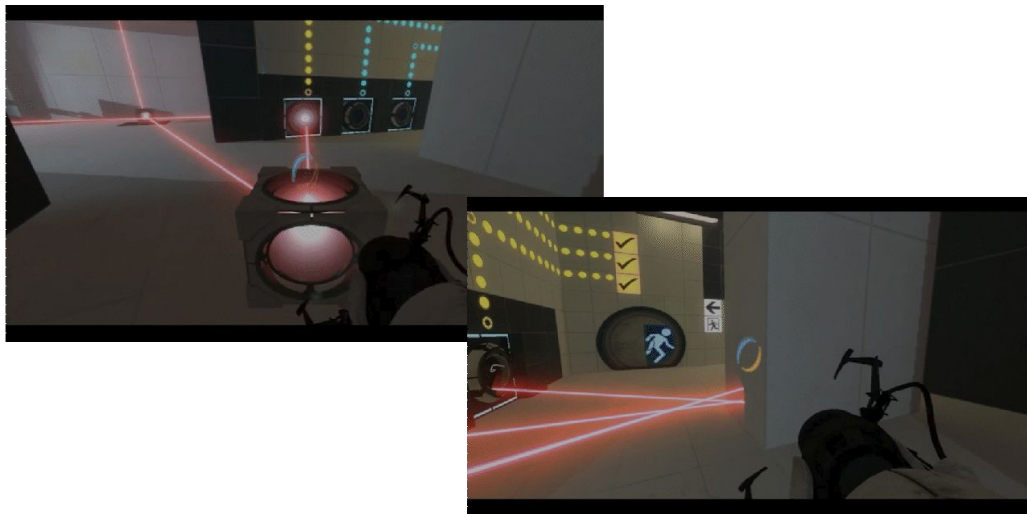
9-3. 레이저 상호작용

레이저와 Door & Button 상호작용

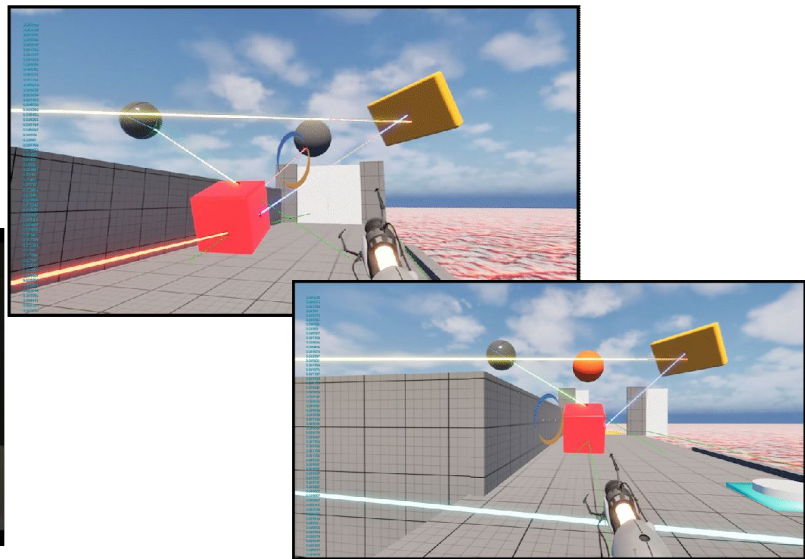
(동적 제어와 가변 시간성 (External Control & Timers))

“버튼 입력 → 레이저 ON/OFF”, “타이머 → 지연된 반응 제어”
“외부반응을 통한 laser 제어와, 타겟과의 상호작용”

원본(원)



구현물



9-4. 레이저 상호작용

레이저와 timer 큐브& filter 큐브간의 상호작용

빛의 파장 λ

R = ●
G = ▲
B = ■

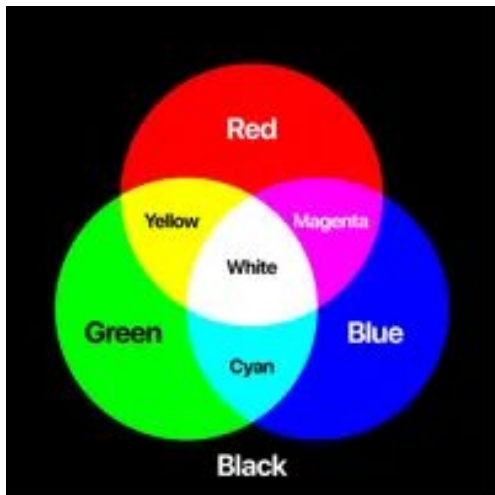
● + ▲ + ■ 을 하여 빛의 파장을 계산하여
퍼즐구현

내가 원하지 않는 색은 filter 하여 작용하지 않게 할수있다.

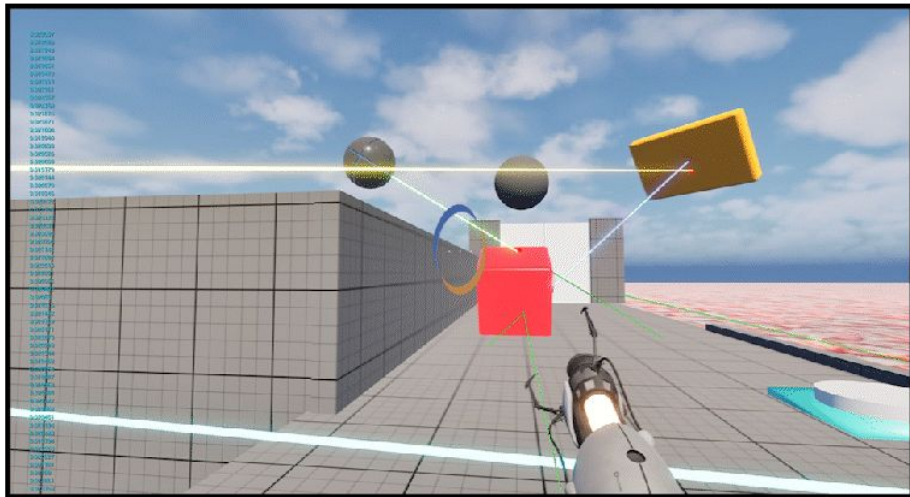
(ex. 빨간색이 들어오면 읽어내지 않는다 -> 조건 불충족으로 무반응)

- 조건을 구성하여 빛의 파장을 합쳤을때 충족하면 작동한다.
- Alpha 를 통해 빛의 세기를 조절한다.

가산 혼합



구현물

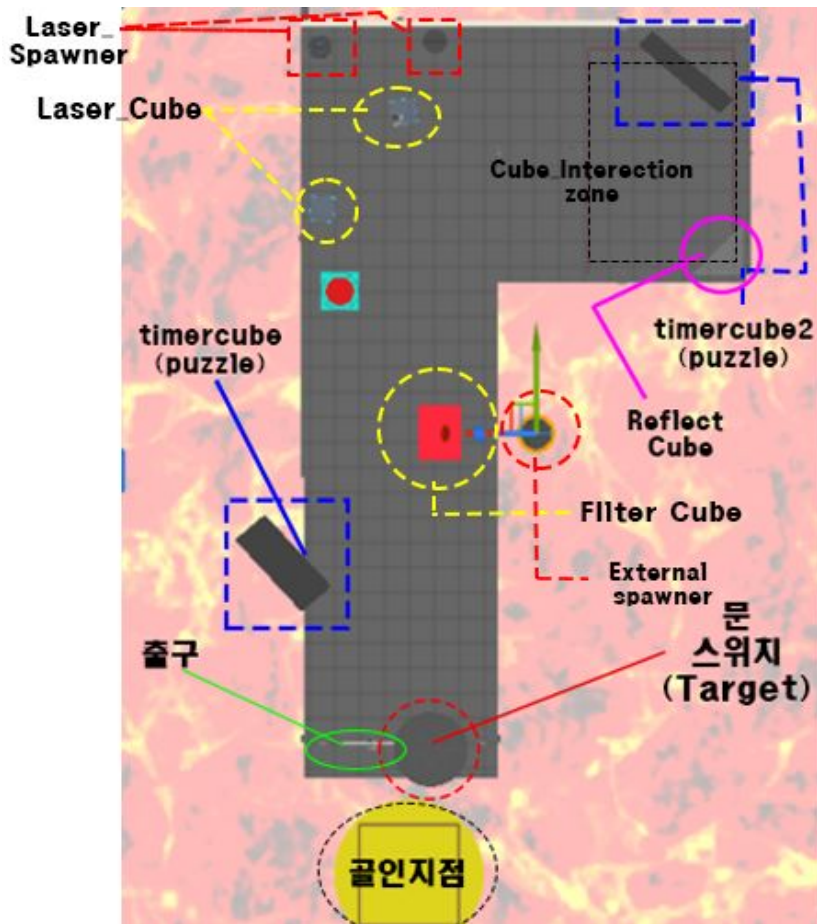


10-1. 퍼즐 연구

- 맵 구성에 따른 레이저 퍼즐연구

1. Laser Spawner 에서 각각 마젠타, Red 레이저를 방출하고있다.
2. Laser Cube중 움직이는 cube 를 이용해 reflect cube에 반사시킨다.
3. 반사된 laser가 timer cube 2 를 활성화 시킨다.
4. timer cube가 timer cube 2의 레이저를 받아 blue 레이저를 내보낸다.
5. switch 를 통해 날라온 cube 가 interaction zone에 들어오면 external spawner 가 활성화 되어 Green 레이저가 출력된다.
6. Filter cube가 Red. Blue . Green 레이저를 받아 조건을 달성하여 하얀색 laser 가 출력된다.
7. 문 스위치에 white laser 가 닿으면 스위치가 활성화되어 문이 열린다.

현재 테스트 도면도



10-2. 맵디자인 연구

- 맵구성과 게임분위기에 맞는 맵디자인연구

원본(원)



어둡고 다소 지저분한.
아포칼립스 느낌에서



GLADOS 나 옆에 살 거고, 80년 후에 죽을 거라는 거 잘 알고 있지?
뒤늦었지만 너한테 생일 선물을 준비하고 있어. 정확히 말하면
뒤늦은 생일 의학 시술이라고 해야겠지. 기술적인 용어로는
임상실험이라고 하지. 하지만 중요한 건 그게 선물이라는 거야.

GameMeca.com

계획

- 멀티 플레이어럼 좀 더 "모던하고 깔끔하게!"
- 화이트 베이스 + 블랙/블루 포인트 = 차갑고 정적인 실험실 분위기



=>

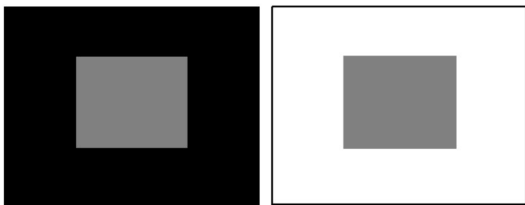
밝고 깔끔하지만,
어둡고 서늘한느낌
(공포감, 긴장감조성)



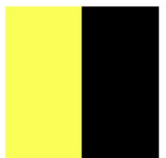
10-3. 맵디자인 연구

- 예상 디자인

- 명도대비를 이용 -> 밝은건 더밝게 어두운건 더어둡게 해서 정적이고 차가운느낌 조성. 또렷한느낌



- 고저 명도를 이용한 배색과 고채도에 의한 배색을 활용해 레이저나 포탈이 좀더 잘보이게.



고저 명도



고채도 배색

=> 명시성 향상



"밝고 어두운 대비와 고채도 포인트 색을 활용해 차갑고 정적인 실험실 분위기를 만들고, 레이저·포탈의 가시성을 높인다."

11. 배주백 담당 구현

프로젝트 담당 파트



간트 차트

						4	5	6	7	8	9	10	11	12	13	14	15	16	17	8
No	대분류	소분류	담당자	우선순위	진행도	기획	프로토													
						월	화	수	목	금	토	일	월	화	수	목	금	토	일	
3	구현 기술RND	포탈건 토이 프로젝트C 개발	배주백	상	완료				목											
6	프로젝트 개설	프로젝트 개설 및 공유	배주백	상	완료									월						
27	맵 상호작용	플랫폼 맵 및 기본 플레이어 추가	배주백	상	완료								화							
28	맵 상호작용	엘리베이터 구현	배주백	상	완료														수	
29	맵 상호작용	점프대 구현	배주백	상	완료														화	
30	맵 상호작용	스위치 Blueprint구현	배주백	상	완료															
31	맵 상호작용	스위치 C++구현	배주백	상	완료														화	
32	맵 상호작용	스위치 상태에 따라서 맵 제어	배주백	상	완료														수	
33	맵 상호작용	터렛 Blueprint구현	배주백	중	완료															
34	맵 상호작용	터렛 C++ 구현	배주백	중	완료															
35	맵 상호작용	큐브 재생성 장치	배주백	하	완료															
36	맵 상호작용	Toggle형 스위치	배주백	하	진행중															
37	맵 상호작용	OneShot 스위치	배주백	하	진행중															
38	맵 상호작용	플레이어 접지면 탐색 기능 구현	배주백	중	진행중															
39	맵 상호작용	포탈2 문 분석	배주백	상	완료															
40	맵 상호작용	스위치 이벤트-델리게이트 시스템	배주백	하	알파진행															
41	맵 제작	발표용 맵 레벨 구성	배주백	중	완료															
45	젤 구현	링크 표현 RND	배주백	상	완료															토
51	프로토 발표	프로토 발표 문서 작성C	배주백	상	진행중															

핵심 기능 구현 내용

라이브러리 구현
스위치 구현
점프대 구현
맵 설계

12. 라이브러리 모듈화

유니티에는 있었는데, 언리얼에는 쓰려고 보니깐 없더라...

Decktris / Assets / 01_Scripts / CoffeeLibrary /

doppleddiggong	속보 개념 추가
Name	Last commit message
..	
00_SampleScene	INIT2
AtlasManager	제거될 타입 표현 추가 및 연출 수정
AuthHelper	INIT2
DebugHelper	INIT2
Editor	INIT2
Extensions	INIT2
GamePrefs	INIT2
MessengerCallBack	속보 개념 추가
ObjectPool	맵 에디터 기본 구성<-> 스테이지 와 연결
Singleton	INIT2
Sound	INIT2
TouchEvent	INIT2
UGUI_Extend	UnitSlot Floaing 연출 겹치는 현상 수정
Util	INIT2
Z_External	컬러 변수들 정리

유니티 때 관리하던 모듈들...



Unity에서 주로 사용하던 패턴, 기능을
Unreal 에서 C++로 구현 및 모듈화

프로젝트 비종속적으로
지속 유지 보수가 가능한 형태로 구현

12. 라이브러리 모듈화

필요하면 만들자.

```
public class Portal2_practice : ModuleRules
{
    public Portal2_practice(ReadOnlyTargetRules Target) : base(Target)
    {
        PCHUsage = PCHUsageMode.UseExplicitOrSharedPCHs;

        PublicDependencyModuleNames.AddRange(new string[] { "Core", "CoreUObject", "Engine", "InputCore", "EnhancedInput", "CoffeeLibrary" });

        PrivateDependencyModuleNames.AddRange(new string[] { });

        // Platform/Public, Player/Public 폴더를 include 경로에 추가
        PublicIncludePaths.AddRange(new string[] {
            Path.Combine(ModuleDirectory, "Platform", "Public"),
            Path.Combine(ModuleDirectory, "Common", "Public"),
        });
    }
}
```

CoffeeLibrary로 모듈 추가

- Portal2_practice
 - Source · 2 프로젝트
 - CoffeeLibrary
 - 종속 요소
 - Private
 - UCoffeeCommonUtil.cpp
 - UEaseComponent.cpp **커스텀 Ease 구현**
 - UObjectPoolManager.cpp **오브젝트 풀 구현**
 - UParabolaComponent.cpp **포물선 구현**
 - Public
 - CoffeeMacro.h **각종 매크로, 주로 싱글톤 관련**
 - FComponentHelper.h **컴포넌트 헬퍼**
 - FEaseHelper.h
 - FMaterialHelper.h **머테리얼 컨트롤 헬퍼**
 - FMathHelper.h
 - UCoffeeCommonUtil.h
 - UEaseComponent.h
 - UEventManager.h **게임 이벤트 <싱글톤>**
 - UObjectPoolManager.h
 - UParabolaComponent.h
 - CoffeeLibrary.cpp
 - CoffeeLibrary.h
 - CoffeeLibrary.Build.cs
 - Portal2_practice
 - 종속 요소
 - Common **이번에 구현한 모듈 구조**
 - Private

12. 라이브러리 모듈화

그래서 뭐가 좋은데?

```
auto AlphaValue :float= LerpAlpha();
// ULOG(Warning, "LerpAlpha : %f", LerpAlpha() );

{
    const auto Color_A = FMaterialHelper::GetVectorParameterValueSafe(MaterialButton, ColorParam);
    const auto Color_B = WarningColor;
    const auto Color_Result = UKismetMathLibrary::LinearColorLerpUsingHSV(Color_A, Color_B, AlphaValue);

    MaterialButton->SetVectorParameterValue(ColorParam, Color_Result);
}

{
    const auto Vector_A = SwitchButton->GetRelativeLocation();
    const auto Vector_B = OriginVector;
    const auto Vector_Result = UKismetMathLibrary::VLerp(Vector_A, Vector_B, AlphaValue);

    SwitchButton->SetRelativeLocation(Vector_Result);
}
```

Material의 특정 값을 가져오거나

```
void AJumpPad::BeginPlay()
{
    Super::BeginPlay();

    this->TriggerDelay = Duration;
    this->ElapsedTime = 0;

    SwitchButton = FComponentHelper::FindComponentByNameRecursive<UStaticMeshComponent>(Owner, this, ComponentNameSWITCH_BUTTON_PATH);
    SwitchCollision = FComponentHelper::FindComponentByNameRecursive<UPrimitiveComponent>(Owner, this, ComponentNameSWITCH_COLLISION_PATH);
}
```

클래스 안의 특정 컴포넌트를 찾아오거나

12. 라이브러리 모듈화

팔각.



```
UObjectPoolManager* PoolManager = GetGameInstance()->GetSubsystem<UObjectPoolManager>();
if (!PoolManager)
{
    UE_LOG(LogTemp, Error, TEXT("AShootingPawn::Fire() failed: UObjectPoolManager subsystem not found.));
    return;
}

// 풀에서 아이템을 가져옵니다.
TSubclassOf<AActor> BulletToSpawn = BulletClass[ShotCount % BulletClass.Num()];
AActor* Actor = PoolManager->GetPoolItem(this, BulletToSpawn);

if (AShootingBullet* ShootingBullet = Cast<AShootingBullet>(Actor))
{
    ShootingBullet->SetOwner(this);
    ShootingBullet->SetInstigator( GetInstigator() );
    ShootingBullet->SetActorLocationAndRotation(FirePoint->GetComponentLocation(), FirePoint->GetComponentRotation());

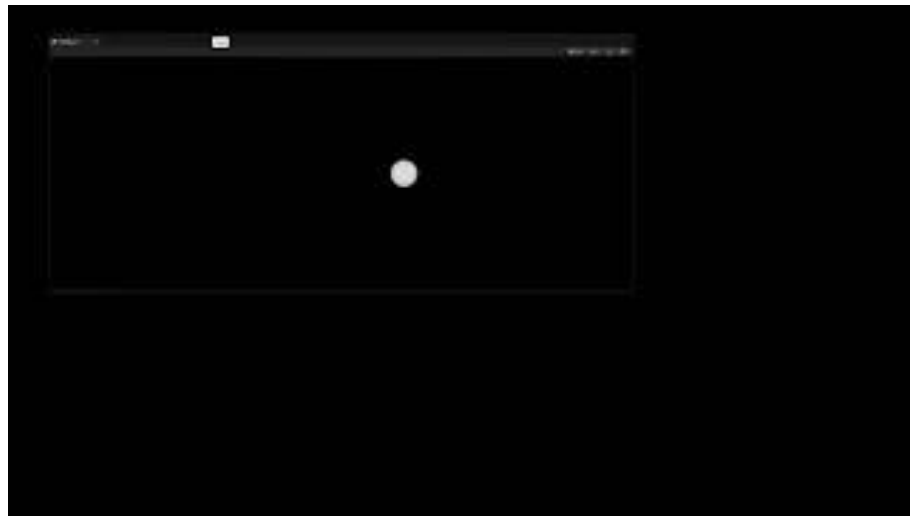
    UGameplayStatics::PlaySound2D(GetWorld(), FireSound);
}
```

GameSubSystem을 이용해서 싱글톤화.

ClassType을 Key로 Key-Value화 해서 오브젝트 풀 매니저 구현

12. 라이브러리 모듈화

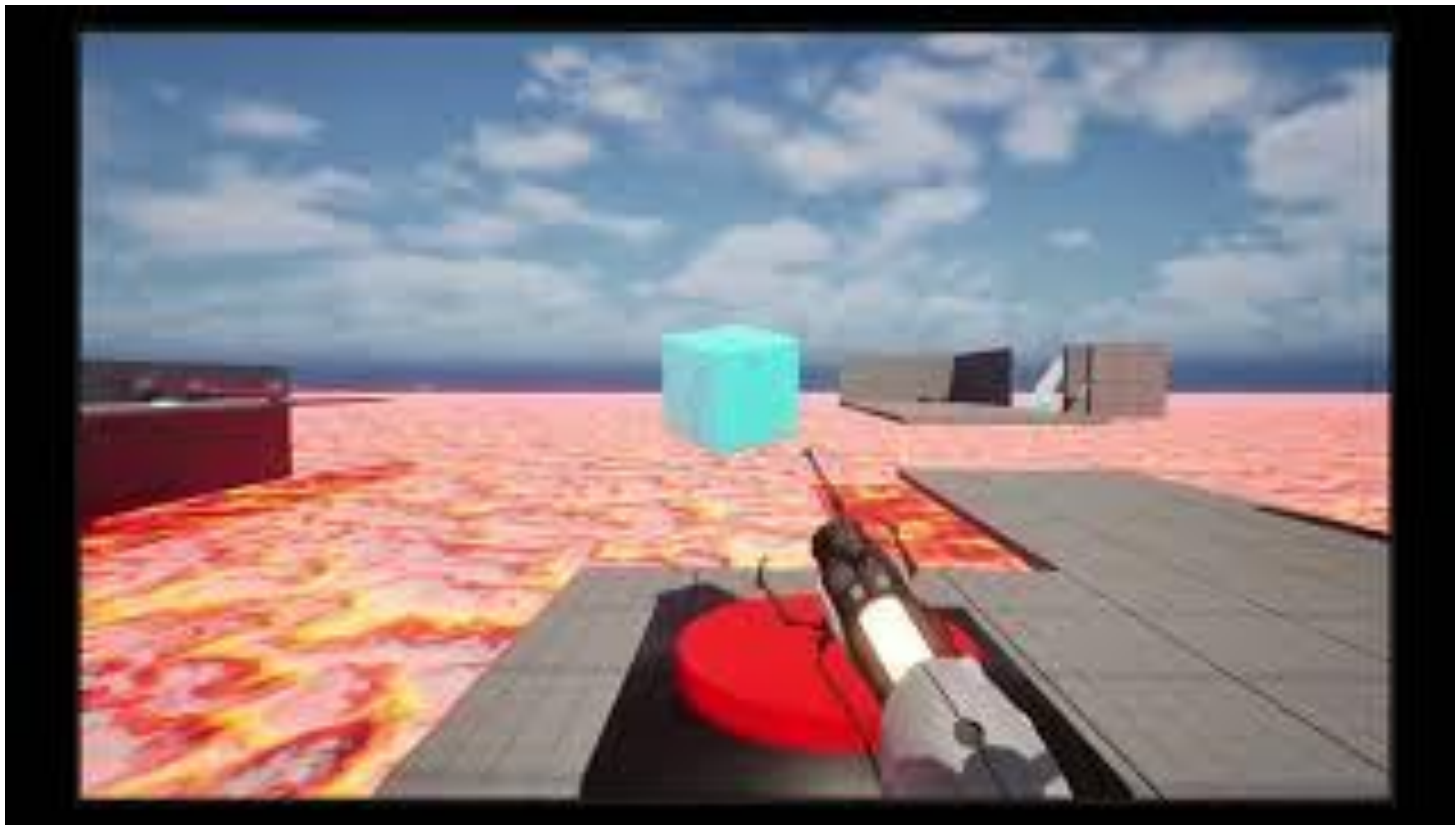
Parabola 와 Ease



언리얼에도 있긴한데, 남 것 말고 내 것

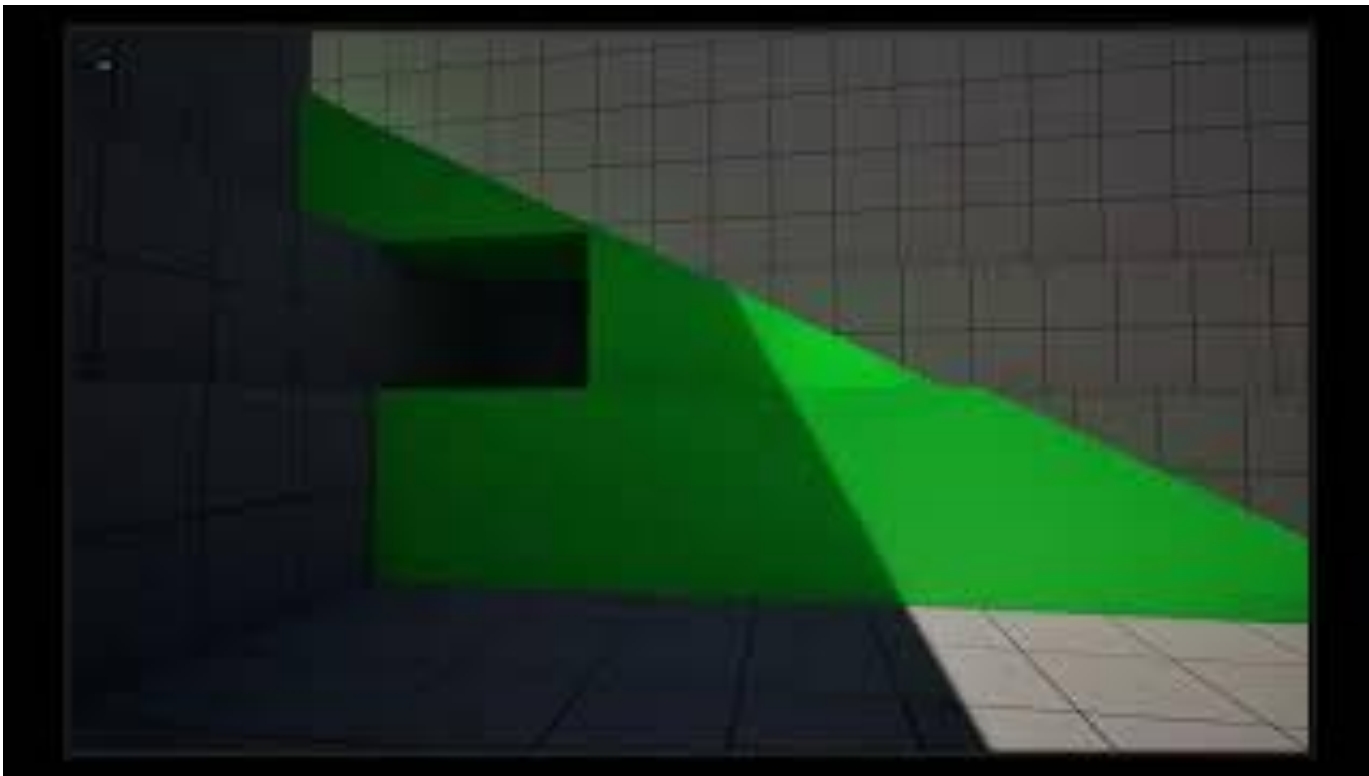
13. 프로젝트에 적용한 것

오늘은, 큐브를 만들고 사람을 날려요



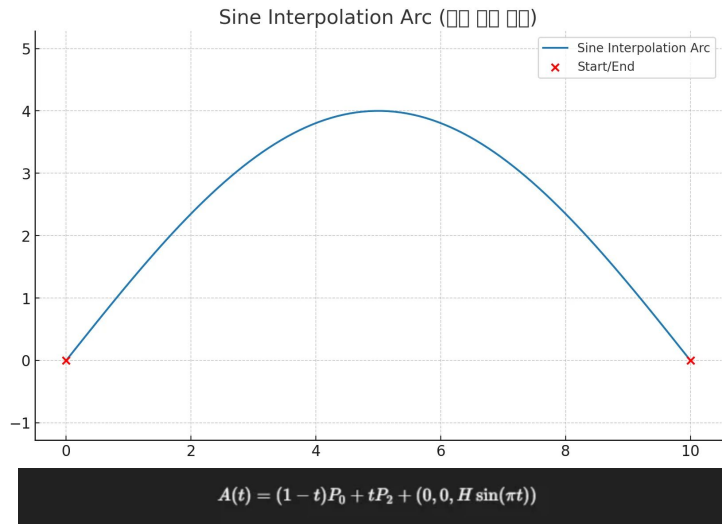
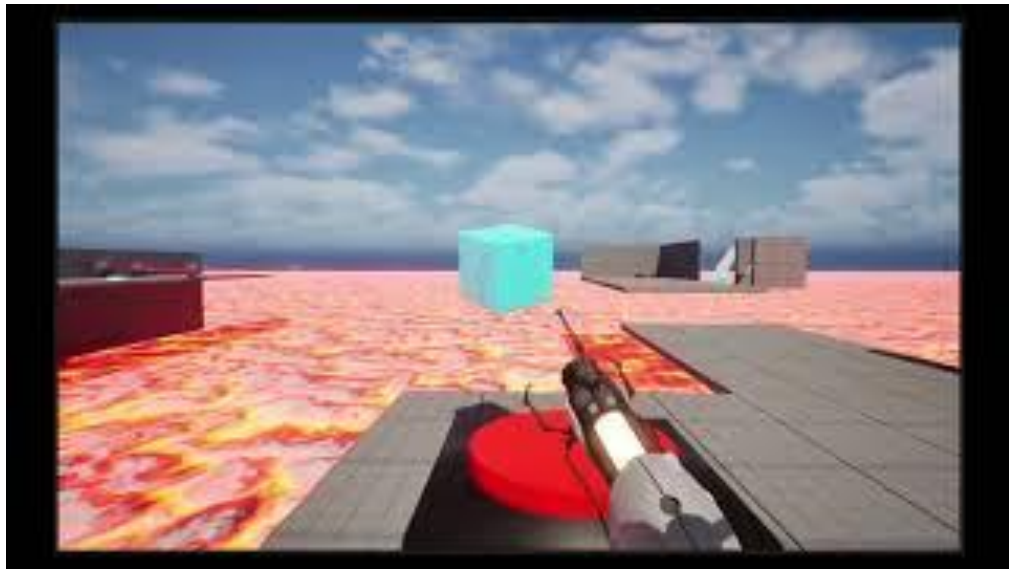
13. 프로젝트에 적용한 것

다음은.... 별 걸 다 할거예요



14. Sine Interpolation Arc

캐릭터와 큐브를 동일한 궤적으로 이동시킨다



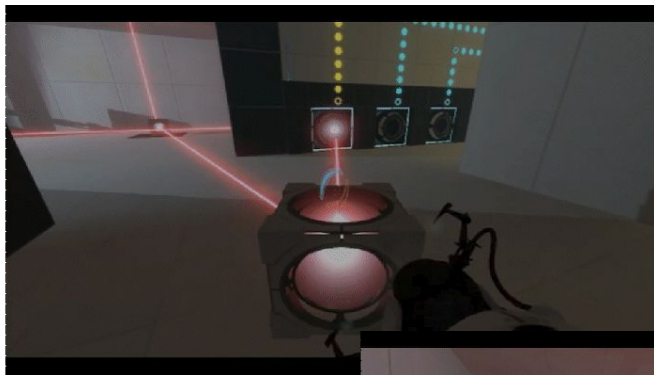
시작점과 끝점을 자연스럽게 연결하는
일관된 아치 궤적이 필요

보간 이동 방식의 한계를 보완하기 위해,
목표 지점에 도달하기 직전에 보간을 종료하고,
순간 속도(velocity)를 계산해 실제 물리에 재적용

15. 알파 구현 계획


앞으로의 진행방향

- 포탈 및 기믹을 활용한 플레이 레벨 디자인
- 포탈과 레이저, 오브젝트 상호작용
- 순차적 오브젝트 제어 연출
- 표현 개선(애니메이션, 맵, 시네마틱, UI 등)





THANK YOU

 포탈2

#3

침착맨도 할 수 있다
FPS어드벤처의 갓겜

