

다른 개발로드맵

프로젝트 목적:

- XR 하이퍼 캐주얼 게임 제작, 마켓(steam, oculus, app store)출시, 스튜디오 드리프트 포트폴리오/브랜딩 (이후 AR 사업 확장 위함, spark ar 및 AR 콘텐츠 기반 숏폼 영상 활용)

프로젝트 배경:

- 기존 멀티플레이어 VR 게임을 개발했으나, 프로젝트 다운사이징 및 vision pro 및 quest pro등 인기 있는 아이템의 파급력도 얻기 위해 스위칭 함

최종목표:

- vision pro / quest pro / vive xr elite 3대 디바이스에 출시 + lidar센서 활용한 하이퍼 캐주얼 게임 포트폴리오

기존 진행하던 VR 프로젝트를 AR 핸드트래킹 기술로 바꾸는 작업

개발 방법 : Oculus Meta Quest 2에서 테스트, 유니티 AR 제작 환경을 이용

- How to Activate Hand Tracking On The Oculus Meta Quest 2
https://www.youtube.com/watch?si=DkcdodbY0Ca0k3DP&v=Yveg_dqBs34&feature=youtu.be

필요 기술

➤ Hand Tracking with Unity XR Interaction Toolkit (주 사용 기술)

: Unity XR Interaction Toolkit 을 이용한 핸드 트래킹은 Unity를 사용해 VR과 AR 프로젝트에서 사용자의 손을 실시간으로 추적하고 상호작용하는 기술. 이 기술은 Unity 엔진의 XR 플랫폼과 연동해 사용자가 손을 사용하여 가상 환경 내에서 물체를 선택, 드래그, 놓기 등과 같은 다양한 상호작용을 수행할 수 있게 해줌. 다음과 같은 기능을 제공해 줌.

- 핸드 트래킹 : 사용자의 손을 정확하게 추적해 가상 환경 내에서 실제 손 움직임을 반영
- 인터랙션 : 사용자의 손을 사용해 가상 물체를 터치, 잡기, 던지기 등과 같은 상호작용을 가능하게 함
- 이벤트 시스템 : 사용자의 상호작용에 반응해 게임 논리를 실행하고 상호작용 결과를 처리
- 다양한 플랫폼 지원 : 다양한 VR 및 AR 플랫폼과 호환성을 제공하므로 여러 종류의 디바이스와 헤드셋에서 사용할 수 있음
- 사용자 정의 기능 : 게임 논리 및 상호작용 스타일을 개발자가 자유롭게 사용자 정의할 수 있는 확장성을 제공

<https://www.youtube.com/watch?v=mJ3fygb9Aw0>

➤ LiDAR(Light Detection and Ranging)

: 라이드는 레이저 관선을 사용해 주변 환경의 거리와 형상을 측정하는 기술. 이 기술은 AR 및 VR과 같은 증강현실 및 가상현실 응용 프로그램에서 중요한 역할을 함. 주로 최신 모바일 기기 및 AR 헤드셋에 통합되어 있으며, AR 게임 개발자에게 강력한 도구로 제공. LiDAR는 주로 다음과 같은 방식으로 AR 게임과 접목될 수 있음.

- 환경 인식 : LiDAR는 주변 환경의 공간 구조를 정확하게 측정할 수 있음. AR 게임에서는 플레이어의 주변 환경을 실시간으로 인식하고, 게임 세계에 이를 반영하는데 사용
- 장애물 감지 및 상호작용 : 주변 객체와의 거리를 정밀하게 측정할 수 있으므로 플레이어가 실제 물체와 상호작용하는 AR 게임을 개발할 때 유용함. 플레이어가 방해물을 피하거나 실제 물체에 가상 물체를 부딪히게 할 수 있음.
- 배경 및 환경 개선 : 데이터를 사용해 주변 환경의 시각적 품질을 향상시키는 데 활용할 수 있음. 게임의 그래픽 효과를 확장하거나 가상 캐릭터와 실제 환경을 더욱 자연스럽게 통합하는데 사용 가능.

https://youtu.be/iAF8AxBNWWhs?si=_Kq2PuzLmkwKaASg

➤ Pass-through

: 사용자가 디바이스를 통해 주변 실제 환경을 볼 수 있게 해주는 것을 의미함. 주로 AR 및 VR 헤드셋에서 카메라를 사용해 주변 환경을 촬영하고 표시하는 용도로 활용. 이 기술을 사용해 다음과 같이 게임과 접목시킬 수 있음.

- 실제 환경 통합 : 사용자가 현실 세계를 계속해서 볼 수 있게 해줌. 사용자가 주변 환경을 보면서 게임의 가상 캐릭터 또는 물체가 나타나도록 만듦
- 확장된 현실 체험 : 사용자가 실제 공간 내에서 다른 위치로 이동하는 등의 경험 제공 가능
- 인터랙션 : 사용자가 실제 환경의 물체와 상호작용 가능. 게임 내 물리적 객체를 활용하는 요소 추가 가능
- 사용자 안전 및 편의성 : 사용자가 주변 환경을 볼 때 물체와 충돌하지 않도록 경고 기능을 추가하거나 사용자의 편의성을 위해 정보를 시각적으로 제공

https://youtu.be/zG5AmL7CeLw?si=7yvMWd51B_1xvtHD

개발 로드맵

1. 핸드트래킹 및 다양한 센서 통합:

- Unity 3D나 Unreal Engine과 같은 게임 엔진을 사용하여 핸드트래킹과 다른 센서(예: LiDAR, Passthrough, Depth Camera)를 통합해야 함
- 각 센서의 SDK(소프트웨어 개발 키트)와 API를 이용하여 데이터 수집과 상호작용 기능을 구현

2. 개발 환경 구축:

- 개발 환경을 설정하고 헤드셋과 센서를 연결하여 개발을 시작
- 필요한 소프트웨어 및 하드웨어 도구를 설치하고 설정

3. 테스트 및 디버깅:

- 실제 하드웨어에서 앱을 테스트하고 디버깅, 시뮬레이터로 테스트하기도 하지만 실제 환경에서의 동작을 확인해야함

4. 사용자 경험 설계 및 개발:

- AR 게임 또는 애플리케이션의 디자인과 사용자 경험을 계획하고 개발
- 핸드트래킹, LiDAR, Passthrough 및 Depth Camera를 사용하여 상호작용 및 가상 콘텐츠 구현

5. 그래픽 및 사운드 디자인:

- 게임의 그래픽 및 3D 모델링, 텍스처, 애니메이션을 개발하거나 구매하여 통합
- 사운드 및 음악 효과를 추가

6. 퍼포먼스 최적화:

- AR 애플리케이션을 위한 최적화 작업을 수행해 안정적인 성능을 유지하도록 함

7. 베타 테스트:

- 베타 터스터 그룹을 모집해 피즈백을 수집하고 버그를 찾아 수정

8. 스토어 등록:

- 앱을 해당 플랫폼의 앱 스토어에 등록하고 승인을 받음

9. 출시 및 마케팅

10. 지속적인 업데이트와 유지보수

Oculus Quest2에서 Unity로 제작한 AR 게임 최적화하기

➤ 렌더링 최적화

- Level of Detail (LOD): LOD 시스템을 활용해 카메라에서 먼 객체를 더 낮은 폴리곤으로 렌더링하고, 가까운 객체에 더 높은 폴리곤 모델을 사용
- 텍스처 압축: 텍스처를 압축해 VRAM 사용량을 줄이고 성능을 향상시킴
- Shader 최적화: 불필요한 계산을 줄이고, 모바일 GPU에 최적화된 셰이더 사용

➤ 코드 최적화

- CPU 및 GPU 프로파일링: Unity의 프로파일러를 사용해 성능 병목 현상을 식별하고 최적화 대상을 파악
- 스크립트 최적화: 불필요한 연산을 줄이고, 데이터 구조 및 알고리즘을 최적화하여 CPU 부하를 줄임

➤ 에셋 최적화

- 불필요한 에셋 제거: 사용되지 않은 에셋을 프로젝트에서 제거하고 메모리 사용량을 최적화
- 에셋 번들 사용: 리소스를 에셋 번들로 묶어서 효율적으로 로드하고 관리

➤ VRAM 관리

- 텍스처 크기 관리: 큰 텍스처를 사용하지 않고 VRAM 사용량을 제한
- 에셋 비트맵 해상도 조절: 게임 실행 중에 동적으로 비트맵 해상도를 조절해 관리

➤ 안정성 및 테스트

- 오류 및 크래쉬 수정: 게임 내 오류와 크래시를 신속하게 수정해 안정성 유지
- Quest2 에서의 테스트: Oculus Quest2 장치에서 게임을 주기적으로 테스트하고 성능 이슈 확인

➤ 애플리케이션 세팅 조정

- 그래픽 설정: 게임 시작 시 그래픽 설정을 플레이어에게 조절할 수 있도록 옵션 제공
- 렌더링 해상도: Quest2의 렌더링 해상도를 조정해 프레임률을 개선

➤ 데이터 비동기 로딩: 큰 에셋을 비동기적으로 로드해 초기 로딩 시간을 최소화하고 유저 경험을 향상시킴

필요한 하드웨어 및 소프트웨어

- Oculus Quest 2 (실제 사용할 AR 테스트 하드웨어)



- 개발용 PC : 강력한 그래픽 카드와 프로세서 탑재 필요)
 - 최소 사양
 - CPU : Intel i5 프로세서 또는 유사한 AMD 프로세서 이상
 - GPU : NVIDIA GeForce GTX 1060 또는 AMD Radeon RX 480 이상
 - 메모리 : 최소 8GB 이상의 시스템 메모리가 필요. 더 많은 RAM 이 개발 작업을 더욱 효율적
- Unity 게임 엔진
 - 유니티 개인용 (Unity Personal): 개인 및 학습 목적으로 무료로 사용 가능. 수익을 창출하지 않는 프로젝트 및 학습용으로 이용 가능하며, 수익 창출을 위한 게임이나 애플리케이션 개발 시 로열티를 지불하지 않아도 됨. 대부분의 기본 엔진 기능을 제공하지만 일부 고급 기능 및 서비스는 프로 버전에만 포함.
 - 유니티 프로페셔널 (Unity Pro): 상업적인 목적으로 사용되는 프로젝트, 회사의 게임 및 애플리케이션 개발에 사용. 프로젝트의 수익과 크기에 따라 라이선스 비용이 부과. 개인용 버전과 함께 고급 기능 및 추가 서비스를 포함하고 있습니다. 이로써 개발자들은 더욱 확장된 기능을 활용 가능. 모든 플랫폼을 지원하며, 고급 기능 및 최적화된 배포 옵션을 제공.

How to get started with Oculus Quest hands tracking SDK in Unity

➤ Oculus SDK(Software Development Kit)

: Oculus VR에서 개발한 가상 현실(VR) 및 증강 현실(AR) 애플리케이션을 개발하기 위한 소프트웨어 도구 모음. Oculus SDK는 Oculus VR 헤드셋 및 기타 관련 하드웨어와 상호작용하고 VR/AR 애플리케이션을 개발하기 위한 API, 라이브러리, 에셋, 플러그인, 및 개발 도구를 제공. 이것은 주로 Unity 및 Unreal Engine과 같은 게임 엔진과 함께 사용되며, Oculus 플랫폼에서 VR/AR 앱을 개발하는 데 필요

- 헤드 트래킹 및 포지션 추적: Oculus SDK는 사용자의 머리 움직임과 위치를 추적하여 3D 가상 환경 내에서 현실감 있는 헤드 모션을 구현하는 데 도움
- 컨트롤러 지원: Oculus Touch 및 기타 Oculus 컨트롤러를 사용하여 손동작을 추적하고 게임 및 애플리케이션에 인터랙티브한 기능을 추가할 수 있음
- 사운드 및 음향: Oculus SDK는 3D 오디오 및 헤드 트래킹 오디오 기능을 제공하여 사용자의 위치와 방향에 따라 사운드가 변경되는 현실적인 음향 경험을 제공
- 플러그인 및 툴킷: Oculus SDK는 Unity 및 Unreal Engine과 통합하는 데 사용되는 플러그인을 포함하여 다양한 개발 도구와 툴킷을 제공
- Oculus Store 연동: Oculus SDK는 Oculus Store와 연동하여 앱을 게시하고 판매할 수 있도록 지원
- 테스트 및 디버깅: Oculus SDK는 애플리케이션을 테스트하고 디버깅하기 위한 도구를 제공하여 개발자가 앱의 성능을 최적화하고 오류를 해결할 수 있도록 도와줌

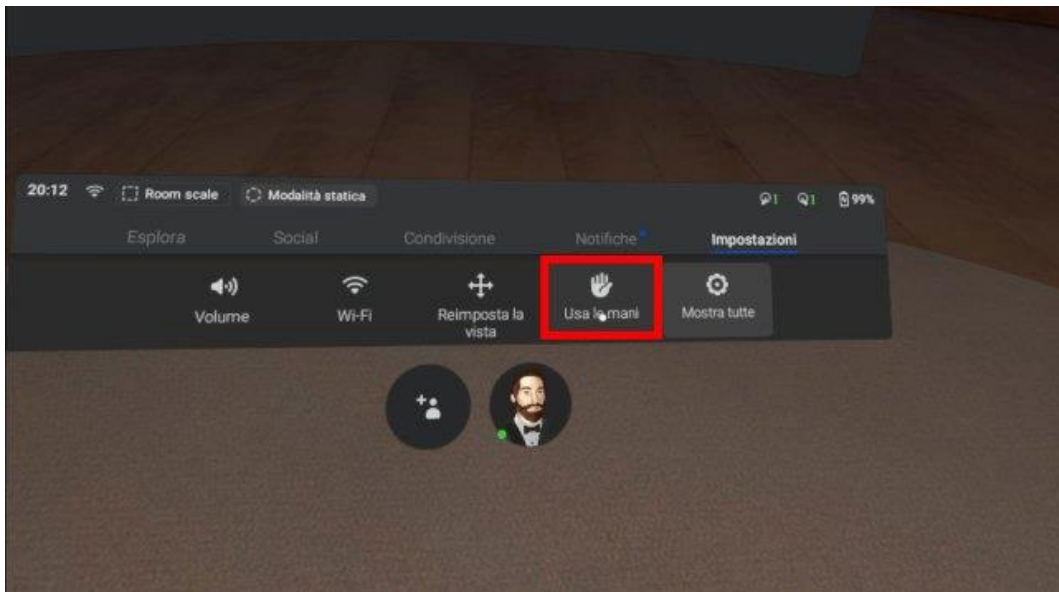
➤ Oculus SDK 문서에서 권장하는 표준 방법을 사용하여 Unity 애플리케이션에 손을 구현

- 샘플 씬 열기> 메인 카메라 제거> OVRCameraRig 추가> 트래킹 원점 설정
- 손 추적 설정: OVRCameraRig의 인스펙터에서 Input 섹션 내의 Hand Tracking Support 드롭다운 메뉴를 찾아 "Controllers and Hands"로 설정. Oculus는 "Hands Only"를 선택하면 앱이 현재 스토어에서 거부될 수 있다고 설명.
- 손 프리팹 추가 (왼손): OVRCameraRig 게임 오브젝트를 확장하고, 그 자식 오브젝트인 "LeftHandAnchor"를 확장. 여기에 Oculus\Assets\VR\Prefabs에서 찾은 OVRHandPrefab 프리팹을 LeftHandAnchor의 자식으로 추가. 이 프리팹을 선택하고 인스펙터에서 OVR Skeleton 스크립트를 찾아 "Enable Physics Capsules" 옵션을 활성화합니다. 이렇게 하면 손의 각 본에 캡슐 콜라이더와 리지드바디가 추가되어 다른 객체와의 상호작용을 감지할 수 있음. 오른손도 Right Hand로 바꾸면 같음.

➤ Oculus SampleFramework 상호작용 시스템을 사용하여 Unity 씬을 만들

- 새로운 씬 생성> 메인 카메라 제거> OVRCameraRig 추가> 트래킹 원점 설정
- 손 추적 설정> 손 프리팹 추가 (위와 동일)
- InteractableToolsSDKDriver 프리팹 추가:
Assets\Oculus\SampleFramework\Core\HandsInteractions\Prefabs\InteractableToolsSDKDriver 프리팹도 씬에 추가. 이것은 더 쉬운 상호작용 엔진을 제공하는 중요한 구성 요소
- 추가한 게임 오브젝트 설정: 방금 추가한 게임 오브젝트를 선택하고, 인스펙터에서 InteractableToolsCreator 동작을 찾아 Left Hand 및 Right Hand에 하나의 도구만 남기고 해당 도구를 FingerTipPokeToolIndex로 설정. 이 도구는 양손의 검지 손가락에 부착되어 특정 객체와 상호작용할 수 있게 해줌.
- 이렇게 하면 Oculus SampleFramework 상호작용 시스템을 사용하여 Unity 씬을 만들게 됨. 이 시스템은 더 고수준이며 사용하기 쉽기 때문에 손의 상호작용을 구현하는데 유용.

➤ Oculus Quest에서 손을 사용하는 방법



애플비전 프로를 겨냥한 방향성 대비책

- Unity 업그레이드: Unity 엔진을 현재 버전에서 더 최신 버전인 Unity 2022 또는 그 이후 버전으로 업그레이드하는 것이 좋음. 새로운 Unity 버전은 개발자들에게 최신 기술 및 개선된 성능을 제공하며, 향상된 기능을 활용할 수 있음
- Shader Graph로의 셰이더 변환: 현재 사용 중인 셰이더 코드를 Shader Graph로 변환하는 것이 유용할 수 있음. Shader Graph는 시각적으로 셰이더를 작성할 수 있는 도구로, 셰이더 개발 및 조정을 더 쉽게 만들어줌.
- Universal Render Pipeline (URP) 채택: Universal Render Pipeline은 Unity의 경량 그래픽 렌더링 파이프라인 중 하나로, 크로스 플랫폼 프로젝트에 적합. URP를 사용하면 더 빠른 렌더링과 더 나은 시각적 품질을 얻을 수 있으며, 모바일 및 VR 플랫폼에서 더 효율적으로 실행됨.
- Input System 패키지 채택: Unity의 Input System 패키지를 사용하여 입력 처리를 최적화할 수 있음. 이 패키지는 다양한 입력 디바이스와 상호작용하는 데 도움을 주며, 복잡한 입력 로직을 더 쉽게 관리할 수 있음.
- Volumes 고려: Unity Volumes는 시네머신 및 포스트 프로세스 이펙트를 쉽게 조작하고 적용할 수 있는 방법을 제공. 이것을 사용하여 게임의 시각적 효과를 향상시키고, 특정 상황 또는 영역에 대한 효과를 쉽게 조절할 수 있음.

