mobilePhoneMISHeart\_SubHeart 프로그램 관련 요약문

1. 사용자 인터페이스

(1) 기본적으로 Landscape Mode로 고정되어 있음

(2) 왼쪽에 위치한 여섯 개의 버튼으로 여섯 종류의 심장 파트를 조정할 수 있음

(ㄱ) 짧은 터치로 해당 심장 파트의 Visibility를 조절할 수 있음

(ㄴ) **(절단 모드)** 긴 터치로 해당 심장 파트에 Cutting Plane을 출현시킬 수 있음

-> 손가락을 뗄 떼 작동하나, ButtonPressDetection.cs의 OnPointerUp()에서 일정 시간 이후 강제로 메시지를 보내어 동작하게 할 수 있음. 이 때 ButtonClickEventHandler.cs의 ButtonPressDetectionManager()도 동시에 수정해야 함.

(ㄷ) **(절단 모드)** 절단 모드 상태에서 다른 버튼을 길게 터치하면 이전 파트의 Cutting Plane은 사라지고 (잘린 단면은 그대로 남아있음) 대상 버튼에 해당하는 심장 파트에 절단면이 생성됨.

(ㄹ) **(절단 모드)** 절단 모드 상태에서 현재 파트 버튼을 길게 터치하면 절단모드에서 해제됨.

(3) 버튼을 제외한 영역에서 한 손가락 터치로 심장 모델을 회전시킬 수 있으며, 화면의 해상도에 기반하여 한 쪽 끝에서 끝까지의 슬라이딩 액션은 약 90도의 회전을 의미함.

(4) 버튼을 제외한 영역에서 두 손가락 터치로 심장 모델을 확대/축소할 수 있으며, 손가락 사이의 거리에 대응하여 선형적으로 반응함.

2. 프로그램 구조

(1) 심장 모델

심장 모델은 Full Objects는 사용하지 않았으며 Part 모델만을 사용함. 해당 모델들이 모두 원점을 기준으로 형성되어 있지 않기 때문에 Renderer의 Bounds (충돌 감지 박스)의 중심점 위치를 기반으로 하여 상위 GameObject인 Human Heart의 위치를 Bounds만큼 이동시켜 전체 모델들을 World Origin 기준점으로 배치시킴. Human Heart 밑에 Partial Model 구조체를 두고, 이 아래 각 심장 파트 모델을 위치시킴.

(ㄱ) 각 심장 모델의 최하단 오브젝트에 customMeshRenderer 스크립트 파일을 만들고, CrossCuttingShader를 연결하여 절단 모드를 수행함. 각각의 customMeshRenderer는 customPlaneManager를 멤버로 두고 업데이트가 발생할 때마다 이후 이야기할 Visual Plane Model과 동기화시킴.

(ㄴ) 구동 구조

(a) 각 파트는 구동 시 cuttingModeEnabled가 거짓으로 설정되며, 각 파트는 모두 보이는 상태로 렌더링된다.

(b) 긴 버튼 입력으로 인해 절단 모드가 발생하면, 해당 심장 파트의 customMeshRender의 cuttingModeEnabled가 On이 되며 CrossCuttingShader의 Clipping 옵션이 켜진다. cuttingModeEnabled는 이 파트가 사용자 입력에 반응하여 절단면을 조절할 것인지를 나타내는 것이며, Clipping 옵션이 1이면 현재 조절하는 Cutting Plane에 따라 이 파트를 절단한다.

(c) 손가락 세 개가 감지되면 첫 번째 손가락의 x 위치의 변위를 측정하여 이를 평면 모델의 거리값에 대응시키고, 이를 Visual Plane에도 업데이트 시킨다.

(2) 시각적 평면 모델

시각적 평면을 보여주지 않아도 상관없다. 현재는 PlaneManager 밑에 파트 이름 + Plane 형식으로 GameObject를 구성하여 그 밑에 Plane 모델 두 개를 붙인 상태이다. 이는, Plane은 normal 방향으로만 그 표면을 렌더링하기 때문이며, 이를 두 개 붙여 어느 쪽에서 보아도 Plane이 존재하도록 한 것이다.

(ㄱ) 각 파트 이름 + Plane의 GameObject에는 customPlaneManager가 존재하며, 이 GameObject의 transform을 바꾸어 시각적으로 존재하는 평면을 조작한다.

(ㄴ) 이 customPlaneManager의 실제 붙어있는 Plane과는 상관없이, 내부적으로 Plane Instance를 만들어 이를 customMeshRenderer의 Cutting Plane과 동기화시킨다. 이는 Unity에서 생성하는 Plane Object는 Mesh Filter로 간주되기 때문에 Plane으로 가져오는 것이 불가능하여 전체 normal이나 distance 값을 가져오는 것이 불가능하기 때문에, 가상의 Plane 객체를 조절하여 이 값을 동기화시키기 위함이다.

3. 추가 구현할 사항 (선택적)

(1) 현재 절단 모드에 들어가있는 파트의 시각적인 표현 (Mesh 색 변화 및 버튼 색 변화)

(2) 초기화 버튼 추가

(3) 현재 Visual Plane과 실제 Cutting Plane이 잘 맞아 떨어지지 않음. 아마도 각 Plane의 크기를 조절하기 위해 X, Z축의 Scaling이 들어가 있고 이게 Rotation이나 Transition에 영향을 미치지 않았나 싶은데, 가능하다면 이를 맞춰주는 것이 좋을 듯 함.