|  |  |  |  |
| --- | --- | --- | --- |
| MORPHOSIS 개발일지 19차 | | | |
| 기간 | 2019-07-30 ~ 2019-08-05 | 작성자 | 신재욱 |
| 작업 내용 | | | |
| 충돌처리 부분에 테스트 중인 부분 있다. 나중에 확인 바람. <- 가 18차 개발일지에서 넘어옴~  Change Scene을 먼저 하는게 맞겠지?  Change Scene을 위해서 먼저 PSO를 전역에서 벡터로 관리하게 변경하였고 PSO Generator를 만들었다.  또한, ObjectManager와 Scene, Object(와 그걸 상속받은 것들)의 소멸자를 전부 정리하여 작성하고 Release()를 만들었다.  그리고 CScene에서 필요없는 부분들을 지우고 간소화하였다.    예에에전에 Change Scene을 만들 때 제대로 안 되서 미뤘던 기억이 있는데 오늘 원인을 찾아냄.  명령어리스트를 BuildObject()에서 Close() 해주었기 때문.  Reset() 해주고 초기화한 뒤, Close()하고 그걸 Execute 해주었다. 그 아래는 덤.  이렇게 했더니 Scene을 바꾸는 것이 된다. 근데 지금 방식은 Scene을 바꾸면서 모든 리소스를 다시 로드하고, 오브젝트들을 다시 생성하는데 이 때 잠시 끊기는 딜레이가 있다.  지금 객체 수도 얼마 안 되는데 이정도로 끊기면 불안하다. 전역에서 관리하고 있으므로 몇몇 리소스는 그대로 유지해도 될 것 같고, 나중에 여유가 되면 쓰레드를 추가하여 Scene에서 Scene으로 넘어갈 때 전환 효과를 넣어주도록 하자.  추후 필요한 작업: Scene을 초기화할 때, 그 Scene을 설명하는 구조체를 밖에 저장해두던지 해서 그 Scene에서는 어떤걸 어떻게 생성하고 하는 것들을 읽어오고 그걸 Object Manager에게 넘겨주는 식으로 해야 할 것 같다. 어떤 Level을 쓰는지 그런거.  UI 구현을 위한 브랜치를 팠다.  UI는 이전에 구현한 적이 있다. 다만, 그 당시 쓰던 코드는 현재 구조와 맞지 않으므로 작동원리와 개념 정도만 가져와도 충분하다. 그리고 사실 내용이 매우 간단하여 가져올 필요도 없다.  무언가 렌더링을 하려면 HLSL에 데이터를 올려줘야 한다.  UIVertex, UIMesh, UIPso를 만들고, 루트시그니처에 UI를 위한 업로드 버퍼 인덱스를 적어준다.  그 외는 오브젝트를 올릴 때와 동일하다. HLSL에선 UIVertexInput과 Output 구조체를 선언하고 따로 VS와 PS를 만들어준다.  이 참에 메쉬랑 버텍스 정리해야지~  루트 시그니처의 인덱스를 제대로 넣지 않으면 HLSL에 값이 제대로 안 올라가고  그 값을 어떤 쉐이더 함수에서 쓰려고 하면 아무런 오류 문구도 없이 PSO 생성만 실패함.  하나하나 변경해보면서 문제를 찾았다. 우웩          하여간 텍스처 메쉬를 띄울 수 있게 되었다. 참고로 이 아이는    실시간으로 크기 조절이 된다. 야호.  이건 디비전에서 사용하는 방식의 플로팅 UI고 크로스헤어 등을 나타낼 일반 UI도 필요함.    일단 이런 식으로 쓸 수 있다. 맞으면 체력바가 깎인다.  여기까지 해놓고 일단 커밋. | | | |