|  |  |  |  |
| --- | --- | --- | --- |
| 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.  오브젝트 매니저가 씬의 역할을 하고 있다.  아니 이럴거면 오브젝트 매니저 왜 써 걍 씬으로 쓰지.    과거의 신재욱이 뭔가 의도가 있었나보다.  브랜치 파서 Scene이랑 ObjectMng랑 어떻게 잘 정리해보려고 했는데 일단 그럴 필요가 없어보이는걸. 오브젝트 쓰는 부분 정리하고 UI나 마저 만지자. 브랜치 삭제!  UI는 플로팅 말고도 화면 비율에 고정되는 부분이 필요함.  HLSL 코드를 추가하고 그에 맞춰서 PSO를 추가하고  CUI를 굳이 나눌 필요는 없고 관리하는 부분에서 m\_vecFloatingUI와 m\_vecDefaultUI로 나눠서 하자.  많은 일이 있었다…………..  메쉬 생성 시에 모든 점을 (0,0,0)에서 생성. Gf2Size만큼 HLSL에서 점을 이동. Gf2Size는 Size와 Scale을 곱한 값. 많은 테스트들을 통해서 화면의 좌표가 좌상단 (-1, 1)에서 우하단(1, -1)임을 확인. 스크린좌표계가 사용하기 편하니까 좌상단을 (0,0), 우하단을 (1,1)로 값을 전해주면 HLSL에서 알아서 위의 좌표계로 바꿀 수 있게 계산식을 작성.  중간에 막 UI 뜨던 것도 안 뜨고 하는 문제가 있었는데 어떻게 해결했는지 모르겠지만 뭔가 하다가 해결이 됐다…………………. 왜 되지..? 오늘부터 종교를 믿어야.  포토샵에서 DDS 파일을 저장할 때 포맷을 RGB 5.6.5에서 ARGB 1.5.5.5로 변경했다.    알파값을 쓸 수 있어야 하기 때문. 만약 좀 더 부드러운 알파표현을 해야 한다면 ARGB 8.8.8.8이나 하여간 다른걸로 변경하자.    화면 비율에 따라서 나오는데 생각해보니까 UI가 화면 비율을 따라가면 안 된다……  위치면 몰라도………… 크기는 안 된다………………….. 애가 옆으로 퍼지면 어떡해.  일단 여기서 커밋.  다른 게임들은 해상도 키우면 버튼 해상도는 그대로고 위치만 비율에 맞춰서 옮겨지는데  (예를 들어 50 \* 50 버튼이 800\*600이든 1920\*1080이든 크기는 계속 50 \* 50)  따라서 그렇게 변경하였다.    커밋하자!  타이틀 씬에서 보여줄 이미지를 만들었다.  이걸로 일단은 구분 가능.    숫자키 8, 9로 씬 변경을 할 수 있다.  카메라 시점을 변경했다.    스플래툰을 보니까 캐릭터는 가운데 있고 조준점이 화면 정중앙에서 살짝 위로, 오른쪽으로 치우쳐져 있었다. 그러면 총을 쏠 때도 총 위치에서 나가겠네. 좋을 듯.  지금은 조준점을 살짝 위로 올려주기만 했다.  오브젝트 매니저가 씬을 80~90% 정도 대체하고 있기 때문에(…) 어떻게든 변경해야 할 것 같다.  체력바가 0 이하일 때도 막 보이길래 뭔가 했는데 체력이 음수로 내려가면서 그 반대 방향으로 나오던 문제였다. CPlayer의 데미지 입을 때 호출하는 함수를 수정했다.    여기서 뭔가 걸렸다!!!!  하지만 다른 테스트 중이었으므로 제대로 못 잡음……  FloatingUI 개수를 1개에서 5개로 늘렸더니 저기서 계속 걸린다. 힙 손상 체크하는 법이래서 넣어놨는데 그동안 한 번도 안 걸려서 잊고 있었다.  아니 그럼 다른건 왜 터지는거지? 체크 함수를 좀 더 두는게 좋겠다.  일단 힙이 터져도 돌아는 감. (…)    아니 그럼 얘는 뭐야. 얘랑 루트시그니처 생성부랑 왜 문제였지?  예로부터 프로젝트를 새로 파서 옮기면 해결되는 문제가 있었다…..  오늘은 다른거 안 건드리고 저 문제를 잡아볼 것… | | | |