|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **주차** | 6주차 | **기간** | 2020.04.20~ 2020.04.26 | **지도교수** | (서명) |
| 이번주 한일 요약 | 김영완: 씬 전환, 파티클 시스템 추가, 경고 처리  박건호: 사운드 매니저, 리소스 구현  박정만: | | | | |

<상세 수행내용>

-씬 전환-

Start, Looby, Room, Game, End의 씬들이 update 함수에서 특정 조건을 만족할 경우 기존의 씬객체의 메모리를 반 납후 새로운 씬을 생성하게 된다.

처음에는 씬을 생성해 놓고 프로그램을 종료할 때까지 계속 사용할까 생각해봤지만, 파티클 구현으로 인해 사용하는 메모리 양이 점차 늘어날 지도 모른다는 생각과 시작할 때마다 매번 초기화 값이 필요하기 때문에 전환할 때마다 새로 생성하는 것이 좋다고 판단하였다.

-파티클 시스템 구현-

충돌 시 이팩트, 움직일 때 바닥에 발생할 먼지, 아이템을 먹었을 때의 이팩트가 필요하다는 의견이 있어서 급하게 파티클 시스템을 구현하였다.

하나의 이팩트를 이루는 입자들을 계산하고, 그리는 역할을 함.

방식은 구조체 버퍼로 uav, srv, 그리고 uav의 내용을 cpu에서 읽을 수 있도록 하기위해 readback힙의 버퍼가 있음. 처음에 파티클 리스트의 정보는 srv를 upload힙으로 만들어서 담아두고 계산 셰이더에서 srv를 참조하여 파티클 운동을 계산한 결과 값들을 uav에 저장한다. 이때 파티클에 대한 인덱스는 uav, srv동일

계산 후 uav 결과를 readback에 copy -> readback의 결과를 파티클 리스트에 copy-> 마지막으로 다시 파티클 리스트를 srv에 copy함. 이렇게 반복함으로써 매 프레임마다 파티클의 운동을 계산하고 그릴 때는 srv의 정보를 통해 파티클을 인스턴싱한다.

-물리 구현-

물리 부분은 아직도 회의 중, 선형운동은 괜찮으나 회전의 운동이 많이 부족한 모습을 보임.

문제점을 못 찾고 있어서 어려움을 겪고 있다.

-경고 처리-

강제 형변환을 통한 경고들을 처리하기위해 float형태의 입력 값들의 뒤에 f를 추가함.

-사운드 매니저 구현-

우리 게임에 사용할 사운드를 출력하도록 FMOD를 사용하여 사운드 매니저 코드를 작성하였다. 채널에 음악파일을 배열로 저장해서 컨트롤이 가능하도록 작성하였다. 채널을 여러 개 만들면 한번에 여러 소리를 출력이 가능해진다.

-리소스 추가-

월드에 배치하기 위해서 돌 오브젝트를 추가로 제작하였다. 이 돌은 트랙 바깥에도 배치하고

장애물 등으로 사용할 예정.

흙먼지 파티클로 사용할 이미지 추가, 이는 플레이어가 이동 시 발 밑에서 나오도록 할 것이다.

room씬, end씬에 사용할 배경화면 추가. 버튼 이미지를 위한 스프라이트 추가.

Blender 3d로 제작함

--

|  |  |  |  |
| --- | --- | --- | --- |
| **문제점 정리** |  | | |
| **해결방안** |  | | |
| **다음주차** | 7주차 | **다음기간** | 2020.04.27 ~ 2020.05.03 |
| **다음주 할일** | 김영완:. 물리 구현, 속도감을 위한 블러링 구현. 게임 로직 추가  박건호: 기타 리소스들 제작  박정만: | | |
| **지도 교수**  **Comment** |  | | |