|  |
| --- |
| **[ Catch Bear (캐치 베어) ]** |
| [ 1주차 ] 2021. 12. 28 ~ 2022. 1. 1 |
| 12/29 수) 게임 설명문서 제작  12/30 목) Catch Bear 게임 프로젝트를 만들고 장치 초기화 부분을 추가함. 수업때와 다르게 각 장치들을 클래스로 나눠서 초기화함. 이 부분은 Engine 프로젝트로 따로 빼서 관리함. |
| [ 2주차 ] 2022. 1. 2 ~ 2022. 1. 8 |
| 1/2 일) 장치 초기화 코드를 수정함. 루트 시그니처, 메쉬, 쉐이더를 추가해서 삼각형을 띄워봄.  1/3 월) 루트 시그니처 내용 추가, 상수 버퍼 추가. 루트 시그니처에 디스크립터 테이블 추가.  1/4 화) X (생일 약속)  1/5 수) 인덱스 버퍼를 추가하여 상수 버퍼로만 삼각형을 그리던 방법에서 상수 버퍼 + 인덱스 버퍼로 삼각형을 그려봄. 텍스처 매핑 추가했는데 프로젝트 설정을 잘못했는지 링크 에러(kernel32가 없음)가 생김. 3시간동안 해결이 되지 않아 이전 커밋(인덱스 버퍼)로 돌려놓음.  1/6 목) 인덱스 버퍼까지 된 상태에서 텍스처 매핑 코드 다시 추가. 텍스처를 띄우는 것 성공함. Depth stencil buffer 추가함. 아직은 스텐실 버퍼는 사용하지 않고 깊이 버퍼만 사용함. 깊이값을 사용하여 앞, 뒤에 있는 텍스처를 순서대로 출력해봄.  1/7 금) 2주차 회의  1/8 토) 쉼 |
| [ 3주차 ] 2022. 1. 9 ~ 2022. 1. 15 |
| 1/9 일) 새로운 Component 브랜치를 만듦. Input과 Timer 추가 중 Fps가 낮게 나오는 오류 발생, 아직 해결하지 못함  1/10 월) FPS 오류 해결(update가 돌아가는 while문을 잘못 씀). Input과 Timer를 추가, 키보드로 이미지 이동할 수 있음. Material 추가함. 메쉬, 쉐이더, 텍스처를 하나로 묶어서 처리해줌. 동일한 쉐이더(ex 돌)를 가진 물체를 여러 개 만들 수 있음.  1/11 화) 컴포넌트 패턴 추가 – Component, GameObject, MeshRenderer, MonoBehaviour, Transform 클래스 추가함. 작년 3DGP1 강의 복습함(장치 초기화 부분)  1/12 수) Scene, SceneManager 추가함. 이제 객체를 Client 코드에서 생성하지 않고 각 씬에서 만들 수 있음. 게임수학 복습 완료함, 직전 학기에 게임수학 수업과 대부분 겹치는 내용이라 생각보다 일찍 끝남. 남은 시간은 카메라를 공부, 구현해볼 계획  1/13 목) Camera 추가함. 그런데 3DGP1 강의 다시 보고 더 공부해야 할 듯. 아직은 카메라가 키보드 이동으로 움직임(우리 게임은 마우스 움직임으로 에임 조절함). 이부분은 프레임워크가 다 완성되고 수정하는 것이 나을 듯(코드가 계속 바뀜). 내일 회의때 이번주에 한 것들 설명하기 위해 3주차에 한 내용 강의 다시 보고 정리함.  1/14 금) 3주차 회의, 가족모임  1/15 토) 카메라 공부함. 여태까지 프레임워크를 다시 살펴봄, 4주차에 아이패드로 클라이언트 프레임워크 흐름 정리해봐야 할 듯. |
| [ 4주차 ] 2022. 1. 16 ~ 2022. 1. 22 |
| 1/16 일) 최상위 클래스인 Object와 리소스를 관리하는(매니저 같은) Resources 클래스 추가함.  1/17 월) 조명 이론 강의 들음  1/18 화) 조명 구현을 위해 여태까지 프레임워크를 조금씩 수정함. 아직 조명 추가는 안함  1/19 수) 조명 추가, 클라이언트 프레임워크 클래스별로 설명 정리함(엔진쪽 완료)  1/20 목) 노멀매핑 추가, 클라이언트 프레임워크의 모든 클래스 다시 공부, 설명(각 클래스 기능 등) 정리, 클라이언트 프레임워크 디버깅 해보면서 흐름도 정리, 클래스 계층구조 정리  1/21 금) 4주차 회의, 스카이박스 추가중  1/22 토) 스카이박스 추가함 |
| [ 5주차 ] 2022. 1. 23 ~ 2022. 1. 29 |
| 1/23 일) Frustum Culling 추가, 절두체 밖으로 벗어나는 물체는 렌더링 하지 않음  1/24 월) 쿼터니언 공부함, 게임수학 수업 내용이랑 겹치기도 하고 증명 내용이 많아서 설렁설렁봄. 나중에 기억 안나면 강의 다시 보면 될듯. 직교투영 공부중.  1/25 화) 직교투영 추가함. Perspective 카메라 말고 orthographic 카메라로 만들어서 UI만 찍도록 함. Layer를 추가해서 카메라에게 설정해준 레이어에 해당하는 오브젝트들만 찍을 수 있음. Resources(리소스 매니저)에 디폴트 쉐이더들을 만들어주는 함수 CreateDefaultShader() 추가함. 자주 사용하는 쉐이더들을 매번 새로 만들지 않고 미리 만들어놓음.  1/26 수) 개인 일정  1/27 목) 뭔가 많이 못함 ,,,  1/28 금) 렌더 타겟(다중 렌더 타겟) 구현중. Deferred Rendering을 위해 클라이언트 프레임워크 수정중  1/29 토) 렌더 타겟 구현 완료. 기존의 forward 방식의 셰이더 말고도 deferred 방식의 셰이더도 추가함. |
| [ 6주차 ] 2022. 1. 30 ~ 2022. 2. 5 |
| 1/30 일 ~ 2/2 수) 연휴  2/3 목) Deferred Rendering 추가중, 내용이 많아서 오래 걸린다..  2/4 금) Deferred Rendering 추가함. 강의에서는 셰이더 파일을 .fx로 생성했는데, 나는 여태 익숙한 .hlsl로 만들어서 사용했다. 그런데 Diffuse와 Specular를 렌더링하는 텍스처에 계속 잔상이 남아서 왜그런지 이유를 찾는게 오래걸렸다. 셰이더 파일을 .hlsl에서 .fx로 형식을 바꾸니까 잔상이 없어졌다. Compute Shader를 공부하고 있다.  2/5 토) Compute Shader를 추가했다. Graphics 부분을 compute로 다 새로 생성해서 코드가 많이 수정되었다. 렌더타겟부터 강의 시간이 엄청 길어서 하나 보는데도 너무 오래걸린다. 하나를 보면 약간 그날이 끝난듯한 기분이 든다. Particle 부분을 끝내고 메인에 병합하고 지형을 시작해야겠다. |
| [ 7주차 ] 2022. 2. 6 ~ 2022. 2. 12 |
| 2/6 일) Particle System 추가함, Instancing 추가함. 셰이더 파일도 불편하지만 디버깅할 수 있는프로그램인 PIX를 설치했다. 파티클 브랜치를 Main에 병합하고 그림자 할 예정  2/7 월) 그림자를 추가함. 강의 후반부로 갈수록 집중이 안돼서 내일 한번 복습해야할 듯  2/8 화) 테셀레이션 공부함. 아직 지형을 띄우진 않았고 큐브를 와이어프레임으로 띄워서 원리를 살펴봄.  2/9 수) 게임에 필요한 사운드 정리함, 맵 구상함. 지형 공부함  2/10 목) 지형 추가함(테셀레이션 된). 맵 디자인해봄. 금요일 회의 준비함(6,7주 한일 정리)  2/11 금) 7주차 회의 |
| [ 8주차 ] 2022. 2. 13 ~ 2022. 2. 19 |
| 2/12 토) 사운드 리소스 몇개 찾아봄, 코딩은 많이 못함  2/13 일) Picking 추가함. Ray casting 사용, 마우스로 오브젝트 클릭하면 그 오브젝트가 선택됨(출력창에 Hit로 출력함)  2/14 월) 몸이 안좋아서 많이 못함  2/15 화) Mesh 강의 듣고 코드 추가하는데 헤더를 못찾는 오류가 계속 뜸. 고치는 중, 일단 인프런에 질문을 올림  2/16 수) 오류가 해결되지 않아서 강의를 다시 보고 다시 코드를 쳐봤는데 오류가 해결되지 않음. 아직 질문에 답변이 달리지 않음 ,,,  2/17 목) 사운드 리소스 찾음, 일단 Mesh 부분 커밋하고 Picking까지 완료된 상태에서 프레임워크 마무리 작업에 들어갈듯  2/18 금) 8주차 회의  2/19 토) 쉼 |
| [ 9주차 ] 2022. 2. 20 ~ 2022. 2. 26 |
| 2/20 일) 지형 공부함  2/21 월) 스카이박스 리소스 찾고 수정함  2/22 화) 지형 평면으로 수정하고 텍스처 수정했는데 게임 이미지와 맞는 텍스처를 더 찾아봐야 할듯. 테스트 UI들 렌더링 안하도록 수정함  2/23 수) 지형 수정함. 테셀레이션 기능 빼고 텍스처 바꿈  2/24 목)  2/25 금) 9주차 회의  2/26 토) 기숙사 가서 청소하고 짐 정리하고 마트갔다 옴,, |
| [ 10주차 ] 2022. 2. 27 ~ 2022. 3. 5 |
| 2/27 일) 카메라 구조 공부함.  2/28 월) 지형 리소스 찾아서 바꿨는데 아직 맘에들지 않음. Grass texture로 찾으니까 우리 게임에 맞지 않은 너무 디테일한 텍스처들이 나옴. 지형 리소스를 더 찾아봐야 할 듯. 단색으로 제작해서 띄워보니까 게임이 너무 없어보임. 텍스처를 찾아서 해야할 듯. 카메라를 고정할 임시 플레이어(큐브)를 띄움.  3/1 화) 카메라를 플레이어에 붙이려면 플레이어를 어딘가에 저장해두고, 그 플레이어의 transform 정보를 계속 받아와야 한다. 처음엔 카메라의 멤버변수로 플레이어를 뒀는데 플레이어 이동에 따라 카메라가 움직여야 하기 때문에 플레이어 클래스를 새로 만들어야 했다. 지금 클라 계층구조를 보면서 플레이어를 추가중이다.  3/2 ~ 3/9) 기숙사 입주 후 코로나로 코딩은 많이 하지 못했다.  3/10 목) 카메라를 드디어 플레이어에 고정했다. 상속, 타입캐스팅을 활용해서 카메라 스크립트에서 플레이어를 가져오려고 했는데 GameObject와 MonoBehaviour간 상속 같은 관계가 없어서 불가능했다. 그래서 Camera 코드에서 힌트를 얻어 현재 활성화된 씬 안의 오브젝트들을 가져온 다음, 객체들을 순회하면서 객체의 이름이 Player인 객체를 가져오는 방법을 사용했다. 플레이어와의 거리랑 각도만 조절하면 될 듯 하다.  3/11 금) 카메라를 플레이어의 뒷모습이 보이도록 거리 조절했다. 일단 플레이어보다 z값을 300.f만큼 빼주었다. 그런데 플레이어를 따라 카메라가 움직이는걸 보니 툭툭 끊기듯이 움직이는 오류가 발생했다. 원인을 찾아봐야겠다.  3/12 토) 쉬었다  3/13 일) 카메라 이동 코드를 이것저것 바꿔보면서 오류를 수정해보려고 했는데 아직 원인을 찾지 못했다.. 내일 다시 찾아봐야겠다.  3/14 월) 드디어 원인을 찾았다. SceneManager에서 카메라를 먼저 생성한 후에 플레이어를 생성한다. 그러면 씬의 멤버변수인 \_gameObjects(vector)에는 카메라가 플레이어보다 먼저 들어가게 된다. 엔진에서 호출되는 게임의 흐름은 현재 씬에 있는 오브젝트들을 Update한 후 Render해준다. 그런데 플레이어가 이동할 때 시간값(델타값)을 곱해준다. 이 값은 매 프레임마다 다르다. 그러니 카메라의 Update() 함수에서 플레이어의 이동량과 플레이어의 Update 함수에서의 플레이어의 이동량이 매 프레임마다 일치해야 같이 움직이는 것처럼 보이는데, 그게 매번 다르다보니 플레이어가 왔다갔다 움직이는 것처럼 보이는 것이었다. 원인을 알아서 일단은 플레이어를 카메라보다 먼저 생성해주니 자연스럽게 움직인다. 내일은 카메라에서 플레이어를 받아와 카메라를 움직이지 말고, 플레이어가 움직일 때(LateUpdate() 에서 호출되는 KeyCheck() 함수에서) 카메라를 바로 움직이도록 코드를 수정해야겠다.  3/15 화) 원래는 카메라의 LateUpdate() 함수에서 현재 활성화된 씬을 불러온 후, 그 씬 안에서 플레이어를 가져와 그 플레이어의 위치를 카메라의 위치로 설정해줬다. 지금은 CameraScript 클래스 안에 FollowPlayer() 함수를 새로 만들어줬다(플레이어를 인자로 받아와 그 플레이어의 위치 정보를 자신의 위치 정보로 설정)  그리고 Player 클래스의 KeyCheck() 함수에서 플레이어가 이동하면 바로 카메라의 위치 정보도 바꿔주도록 수정했다. 처음에는 player도 씬 안에서 활성화된 객체들을 순회하면서 가져왔는데, 이번에 코드를 보면서 컴포넌트는 자신의 주인을 알고 있고 게임 오브젝트는 자신이 가지고 있는 컴포넌트들을 알고 있는 것을 기억해냈다. 컴포넌트는 GetGameObject() 함수를 실행하면 자신의 주인인 게임 오브젝트를 가져올 수 있고, 게임 오브젝트는 자신이 가지고 있는 컴포넌트와 스크립트를 멤버변수로 가지고 있다. 그래서 현재 씬 안에 있는 카메라만 순회하면서 가져왔다. 그 카메라를(GameObject형) CameraScript로 타입캐스팅 해서 CameraScript::FollowPlayer() 함수를 실행해줬더니 어제 임시로 고쳤을 때와 똑같이 출력되었다. 여기서 게임 오브젝트가 가지고 있는 스크립트(MonoBehaviour형)를 가져올 수 있도록 GetScripts()와 GetScript() 함수를 만들어줬다. GetScripts()는 벡터를 반환해주고, GetScript()는 인자로 넘어온 인트값에 해당하는 인덱스의 스크립트를 반환해주는 함수이다. |