컴퓨터 그래픽스 최종 프로젝트 제안서 2019184015 서정민 // 2021184018 송영준

1. 프로젝트 소개

저희가 제작한 프로젝트는 3d 1인칭 슈팅게임입니다.   
플레이어 주변으로 계속 등장하는 적을 처치하며 버티는 게임입니다.

1. 프로젝트 구조

• main.cpp  
입출력과 타이머를 처리합니다. 기본적으로 키보드 입력으로 플레이어 조종과, 마우스 입력으로 화면 전환을 처리합니다.  
타이머에서는 조명의 계산으로 낮과 밤을 표현해 보았고, 플레이어와 같은 객체들의 업데이트와 충돌처리를 담당합니다.  
카메라는 플레이어 좌표를 기준으로 구 운동을 합니다.  
  
• building.cpp

Typedef struct Building{

GLfloat x, y, z;  
 Vertex scale;

}

building에서는 건물 초기화, 건물 그리기와 플레이어, 적, 총알과의 충돌처리를 합니다.

• bullet.cpp

typedef struct Bullet {

GLfloat x, y, z;

GLfloat dx, dy, dz;

int damage;

bool damaged;

};  
bullet에서는 총알 발사, 총알 그리기, 총알 이동하기, 총알과 적의 충돌처리를 합니다.

• collision.cpp  
collision에는 충돌처리를 보조하는 원 충돌처리 함수와 사각형 충돌처리 함수가 있습니다.

• enemy.cpp  
typedef struct Enemy {

GLfloat x, y, z;

GLfloat dx, dy, dz;

GLfloat angleX, angleY, angleZ;

GLfloat speed;

GLint hp;

GLint damage;

bool damaged;

int backframe;

};  
enemy에서는 적 초기화, 적 이동, 적 그리기, Astar알고리즘을 참고해서 만든 이동방향을 결정하는 AStar, 사망한 적이 쓰러지도록 하는 함수가 있습니다.

• floor.cpp  
floor는 화면에 그릴 바닥의 vertice와 indices들을 정의하고 그리는 역할을 합니다.

• hexahedron.cpp  
hexahedron에서는 육면체의 vertice와 indices를 정의합니다.

• particle.cpp  
struct Particle {

glm::vec3 position;

glm::vec3 velocity;

glm::vec4 color;

int life;

};  
particle에서는 구들을 각자 다른 방향으로 이동시킴으로써 폭죽과 같은 터지는 이펙트를 만들어냅니다. 파티클을 만들고, 파티클을 움직이게하고, 파티클들을 그립니다.

• player.cpp  
typedef struct Player {

GLfloat x, y, z;

GLfloat dx, dy, dz;

GLfloat angleXZ;

GLfloat angleY;

bool gun;

vec3 gunMuzzleWorldPositionVec3; // 총의 위치를 저장할 변수

};  
플레이어에서는 플레이어를 초기화하고, 플레이어를 그리고, 점프와 점프를 현실적으로 표현하기 위해 중력을 표현했습니다. 입력에 따라 플레이어의 이동과 적과의 충돌, 다음 이동 방향을 계산하는 함수가 있습니다.

• pyramid.cpp  
pyramid에서는 사각뿔의 vertice와 indices들을 정의합니다.

• randoms.cpp  
randoms는 랜덤한 번호를 시드값에 따라 발생시키는 역할을 합니다.

• readObj.cpp  
readObj에서는 교재에서의 obj파일을 읽는 코드에 더해 색상과 노말까지 읽도록 하였습니다.

• rwTile.cpp  
rwTile 에서는 타일을 만들어서 그 타일에 빌딩을 세우고 그 타일을 기반으로 빌딩과의 상호작용을 처리하는 역할을 합니다.

• shader.cpp  
shader에서는 교재의 코드로 셰이더를 읽고 만드는 역할을 합니다.

• sphere.cpp  
sphere에서는 구의 vertice와 indices들을 정의합니다.

1. 프로젝트 진행 상황 - 제안서에 작성했던 프로젝트 규모를 사용했습니다.

플레이어 스킬 - 투시경 (건물을 반투명으로 만들어 건물 너머까지 볼 수 있게 함)

맵: 단일 평지 맵, 건물과 같은 오브젝트로 장애물 표현

적: 플레이어 주위에서 일정 주기마다 스폰한다. 플레이어를 계속 추적한다.

적은 머리에 총을 맞았을 때와 몸에 맞았을 때 피격판정이 다르다.  
적은 총알을 머리에 맞으면 즉사하며 파티클을 만들어냅니다. 몸통에 맞으면 뒤로 밀려나가며 체력이 감소됩니다.

피격 범위에 따라 적의 쓰러짐 방향이 달라진다. (X)

무기: 2종류

1. 에임에 맞춰 총알을 발사하는 라이플
2. 에임 방향으로 다수의 총알을 발사하는 샷건
3. 팀원간 작업한 내용

서정민 – 플레이어, 카메라, 조명, 안개, 미니맵, 무기별 총알 발사의 차이, 총알, 문서작업  
송영준 – obj파일 읽기, 적 오브젝트 구현, 맵 구현, 반투명 처리, 건물 개발, 깃허브 관리

1. 스크린샷  
   스크린샷, 보라색이(가) 표시된 사진

   자동 생성된 설명  
   스크린샷, 다채로움, 공이(가) 표시된 사진

   자동 생성된 설명
2. 명령어

Wasd – 이동  
space bar – 점프  
tab key – 무기 변경  
1 – 반투명 효과  
q - 종료  
마우스 작동 – 화면 변환  
마우스 왼쪽 클릭 – 총알 발사

1. 프로젝트 개발 소감 및 후기

서정민 – 2d에서 3d로 넘어오니 계산이 더욱 복잡해져서 힘들었습니다. 삼각함수의 계산과 벡터, 행렬의 계산이 매우 복잡했습니다. 카메라나 조명의 설정으로 보이는 화면이 많이 달라진다는 것을 알 수 있었습니다.  
  
송영준 - 3d 공간상에서의 버그가 2d에 비롯해 기하급수적으로 늘어서 버그 잡는 것이 힘들었다. 하지만 2차원상에서보다 더 자유로운 표현이 가능했고, 2d환경에서 개발할 때 보다 버그가 적었던 부분들도 보이는 것 같아 더 즐겁게 했던 것 같다. 팀 프로젝트 개발과정에서 내가 개발한 분야가 더 적었던 부분과, 잡다한 버그를 제대로 잡지 못해 깔끔하지 못한 결과를 낸 것이 아쉬웠다.