

컴퓨터 그래픽스

- OpenGL FPS 게임 구현

ICT 융합공학부

202004069

신민제

목차

1. 프로젝트 개요
2. 시스템 구조
3. 주요 기능 구현
4. 기술적 특징
5. 시연 및 결과
6. 향후 발전 방향



프로젝트 개요

프로젝트 목표

OpenGL을 활용한 FPS 화면 구현

개발 환경



C++



FreeGLUT

구현 내용

1

Camera

FPS 시점 제어 알고리즘

1. 마우스 offset 계산
2. 감도 적용
3. 카메라 회전 각도 업데이트
4. 수직 회전 제한
5. 수평 회전 정규화

2

ParticleSystem

Particle 업데이트 및 렌더링

1. 각 Particle 업데이트
2. 수명 감소
3. 수명 종료 체크
4. 물리 시뮬레이션

3

발사 시스템

Ray - World 충돌 검사

1. Ray 방향 정규화
2. 평면과의 교차점 계산
3. 충돌 검사
4. 교차점 계산
5. 벽 범위 내 충돌 확인

4

연속 발사 시스템

시간 기반 연속 발사 제어

1. 발사 타이머 업데이트
2. 발사 간격 체크
3. 화면 중앙 좌표 계산
4. RayCast 실행
5. 타이머 리셋

구현 내용

5

조명

- 주변광 / 확산광 / 반사광
- 광원 위치 설정
- 재질 속성 관리

6

텍스처 매핑

- 체크무늬 텍스처
- UV 좌표 매핑
- 텍스처 필터링

7

모델링

- 총구(Cylinder로 표현) 구현
- 위치 / 회전 변환
- 재질 설정

8

입력 처리

마우스 입력

- 좌클릭
- 화면 회전
- 마우스 감도 적용

키보드 입력

- 프로그램 종료 (Q / ESC)
- 셰이딩 모드 (S)
- 와이어 프레임 전환 (W)

헤더

1

GlobalVariables.h

- 전역 변수 목적
 - cameraRotX, cameraRotY: FPS 시점 회전 제어
 - cameraX, cameraY, cameraZ: 3D 공간에서 카메라 위치 제어
 - 모든 소스 파일에서 접근 가능한 카메라 제어 시스템 구현

2

Vector3.h

- 구조체 멤버 변수
- x, y, z: 3차원 벡터 좌표값 저장

3

ParticleSystem.h

- public 메서드
 - ParticleSystem(): 파티클 시스템 초기화
 - update(): 파티클 물리 및 생명주기 갱신
 - render(): 파티클 시각화
 - spawnParticles(): 새로운 파티클 생성
- private 메서드
 - generateRandomVelocity(): 랜덤 속도 벡터 생성
 - renderParticle(): 개별 파티클 렌더링

4

RayCaster.h

- public 메서드
 - RayCaster(): 레이캐스터 초기화
 - castRay(): 마우스 위치에서 레이 발사
- private 메서드
 - getWorldRay(): 스크린 좌표를 월드 레이로 변환
 - intersectWall(): 레이-벽 충돌 검사

CPP

1

main.cpp

- 초기화 함수
 - InitLight(): OpenGL 조명 시스템 설정
 - CreateCheckerTexture(): 체커보드 텍스처 생성
 - MyReshape(): 뷰포트 및 투영 설정
- 입력 처리 함수
 - MyMouseMove(): FPS 시점 카메라 회전
 - MyMouse(): 마우스 클릭 처리
 - MyKeyboard(): 키보드 입력 처리
- 렌더링 함수
 - MyDisplay(): 메인 렌더링 루프
 - DrawWall(): 텍스처된 벽 렌더링
 - DrawGun(): FPS 총 모델 렌더링

2

ParticleSystem.cpp

- 파티클 관리 함수
 - update(): 물리 기반 파티클 업데이트
 1. 중력 적용
 2. 수명 관리
 3. 위치 갱신
- 렌더링 함수
 - render(): 전체 파티클 렌더링
 1. OpenGL 상태 관리
 2. 알파 블렌딩
 - renderParticle(): 개별 파티클 시각화

3

RayCaster.cpp

- Ray Cast 함수
 - castRay(): 레이 발사 및 충돌 처리
 - getWorldRay(): 좌표계 변환
 - intersectWall(): 충돌 검사 알고리즘

구현 화면

