

물리적 엔진 프로젝트 보고서

본 프로젝트는 Python 과 Pygame 을 기반으로 한 2D 물리 엔진을 구현하였습니다. 이 엔진은 강체 동역학, 충돌 감지 및 입자 효과를 시뮬레이션하도록 설계되었습니다.

프로젝트 기능

1.GJK 충돌 감지

두 볼록 다각형의 충돌 여부를 감지합니다.

Minkowski 차와 단순형(simplex)을 이용하여 원점을 점진적으로 접근해 충돌을 판단합니다.

2.강체 동역학

강체의 선형 운동과 회전 운동을 시뮬레이션합니다.

중력 및 외력을 지원합니다.

3.탄성 충돌 반응

충돌 발생 시 물리 법칙에 따라 충격량을 계산하고 강체의 속도와 방향을 업데이트합니다.

4.입자 시스템

폭발, 불꽃과 같은 시각 효과를 시뮬레이션합니다.

랜덤 입자를 사용하여 현실적인 물리 현상을 구현합니다.

코드 모듈

config.py

프로젝트의 전역 상수(예: 창 크기, 중력 등)를 정의하는 구성 파일입니다.

objects.py

강체 객체의 속성과 동작(위치, 속도, 질량 등)을 정의합니다.

physics.py

중력, 충돌 감지(GJK), 탄성 충돌 반응을 처리하는 핵심 물리 엔진 모듈입니다.

particles.py

입자 효과를 생성하고 입자의 상태를 업데이트하는 입자 시스템을 구현합니다.

main.py

모든 모듈을 통합하고 엔진을 실행하는 메인 프로그램입니다.

세부 사항 구현

1. GJK 충돌 감지

GJK 알고리즘은 Minkowski 차를 통해 두 물체가 충돌했는지 여부를 원점에 접근하여 판단합니다. 지지 함수를 사용하여 특정 방향의 가장 먼 점을 계산하고, 원점 포함 여부를 판단하기 위해 단순형을 업데이트합니다.

2. 탄성 충돌 반응

충돌 후 충격량을 계산하고 물체의 질량에 따라 속도와 방향을 업데이트합니다. 법선 벡터를 통해 상대 속도 성분을 분석하여 반발 방향이 물리 법칙에 부합하도록 보장합니다.

3. 입자 효과

입자의 초기 위치와 속도를 무작위로 생성하여 폭발 및 연기 효과를 시뮬레이션합니다. 입자는 수명 동안 점차 감소하고 사라집니다.

프로젝트 구조 및 프로세스

1. 물리 엔진과 입자 시스템 초기화.

2. 메인 프로그램에서 강체 객체를 생성하고 초기 매개변수를 설정.
3. 매 프레임 물리 엔진을 업데이트하여 중력 및 충돌 논리 처리.
4. 충돌 발생 시 입자 효과 트리거.
5. Pygame 을 사용하여 강체와 입자를 렌더링하고 시각적 효과 생성.