



Project #3: Develop Your Own Physics Engine

Pygame framework를 베이스로 나만의 물리 엔진을 제작해봅시다.

완벽한 하나의 물리엔진을 만드는 것이 아니라 물리 엔진의 다양한 세부 요소 중에서 관심이 있고 중요한 기능을 선택하고 개발 및 프로그래밍 경험을 해보는 것이 이번 Term Project의 목표입니다.

첫 번째 주요 평가 요소는 1) 실습 및 수업에서 배운 기능 혹은 알고리즘을 구현하는 것, 2) 해당 내용을 발전시키고 심화하는 것입니다.

두 번째 주요 평가 요소는 본인이 선택한 기능의 물리 엔진으로서 중요성과 발전시킨 방향의 필요성과 의미를 설명하는 것입니다.

제출시에 본인이 구현한 기능을 표현할 수 있는 간단한 데모 프로그램을 작성하여 나만의 물리 엔진의 기능과 코드를 본인의 기여사항 위주로 설명해봅시다.

일정 관련

- Project Consulting: To be determined
(make up class & only for those who want to participate)
- Presentation: 2025.12.09 – 2025.12.11 (3 min/student)
- Due Date: 2025.12.08 23:59:59 -> Upload to e-campus

제출사항

- Submission: Code, Report, Short Execution Video, GitHub Link, Executable File (Link [#1](#) & [#2](#))



2025 Fall SWCON211 Introduction to Game Programming
Department of Software Convergence, Kyung Hee University
TA: Inhyuk Song; Email: cherished990204@khu.ac.kr
TA: Hyunseok Jung; Email: gustjr325@khu.ac.kr
Professor: Seungjae Oh; Email: oreo329@khu.ac.kr

-
- Report Contents:
 - Physics Engine Design & Structure
 - Engine Features with Code Descriptions
 - Technical Implementation & Own Contributions
 - Report Format: PDF -> All the other files in a “single” zip file.

평가지표

- 수업 및 실습에서 다룬 내용과 더 심화된 물리 엔진 관련 기술 중에서 본인이 직접 구현한 내용 및 기여사항
- 엔진 기술 개발 중요성 및 필요성 (30%)
- 엔진 기술 개발 기본 구현 (20%)
- 엔진 기술 개발 심화 구현 (50%)

Q&A

- 기 개발된 외부 물리엔진 라이브러리의 단순 활용은 불가합니다.
- 이미 구현된 기능의 API를 활용하거나 library를 import하여 실행만 하는 것은 지양합니다.
- 즉, 이미 구현된 API나 open source를 활용하여 추가적인 기능을 구현하는 것을 지향합니다.
- 일반적인 용도 및 계산을 위한 python library 혹은 package는 활용 가능합니다.



2025 Fall SWCON211 Introduction to Game Programming
Department of Software Convergence, Kyung Hee University
TA: Inhyuk Song; Email: cherished990204@khu.ac.kr
TA: Hyunseok Jung; Email: gustjr325@khu.ac.kr
Professor: Seungjae Oh; Email: oreo329@khu.ac.kr

구현 기능 예시 (하단 리스트로 제한하지는 않고 더 제안 가능)

- A. Reference Text Book: [Game Engine Architecture](#) & [Real-time Collision Detection](#)
- B. Collision Detection: GJK collision detection, 3D SAT, OBB, Moving Objects, Concave Object, Bounding volume hierarchy, Convex Hull Algorithm, [Optimization](#), ...
- C. Rigid Body Dynamics
- D. [Impulsive Collision Response \(with Torque\)](#)
- E. Particle System & Simulation: fluid, smoke, fire, explosion, ...
- F. [Numerical Methods](#): modified Euler method, RK4, Velocity Verlet, ...
- G. Model Deformation ([Free-form Deformation](#))
- H. Deformable Body