Project: Asteroid field

Physically-Based Simulation 2017

Alexander Lelidis Andreas Emch Uroš Tešić

Goals

Asteroid field simulation in real time

N-Body simulation

- Parallel CPU
- ~~FMM~~

Collision detection

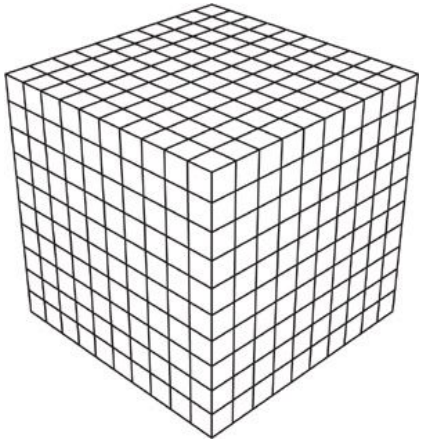
~~Fraction~~

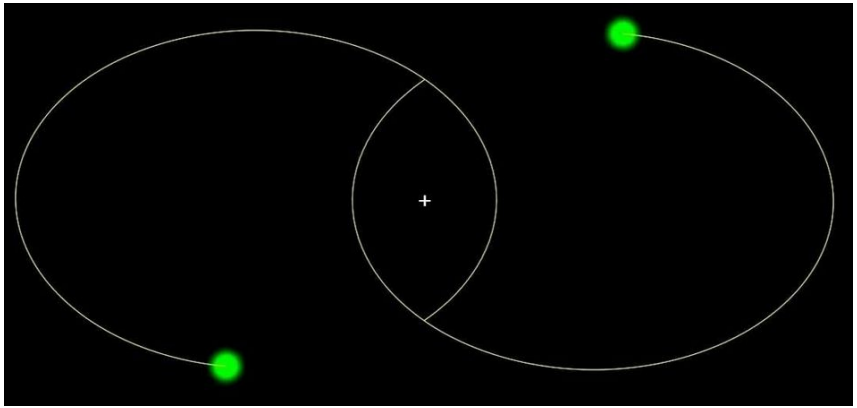
System overview

- OpenSceneGraph:
  - Load and manage the scene and render thread
  - Interaction with OpenGL (textures, shaders, ...)
- Model preparation
  - Convex-Hull
  - Inner moment of inertia
- N-Body manager
  - Calculate interaction forces
  - Simulate objects
- Physic manager
  - Collision detection (broad and narrow phase)

Implementation overview (N-Body)

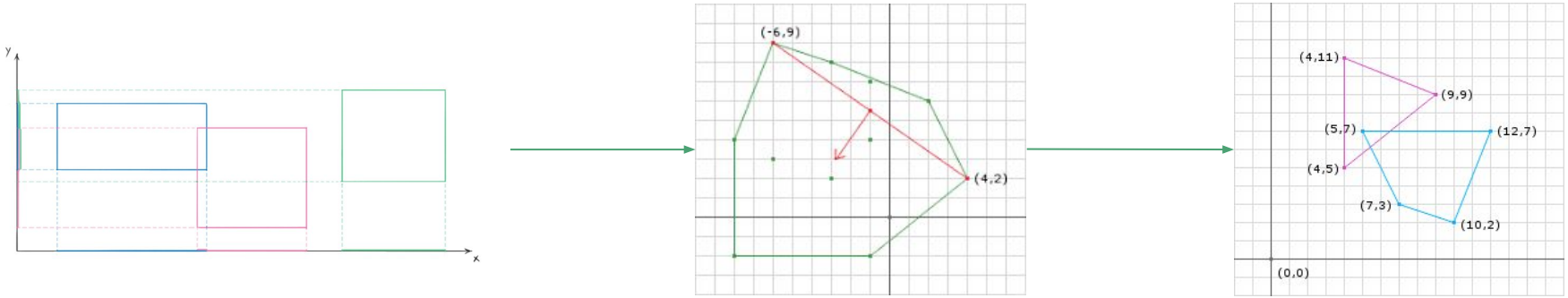
- N-Body
  - Calculations of interaction-forces
    - Uniform spatial grid for speeding up calculations between interactions
    - Interaction-grid-size cut of by threshold
  - Simulate objects with new forces

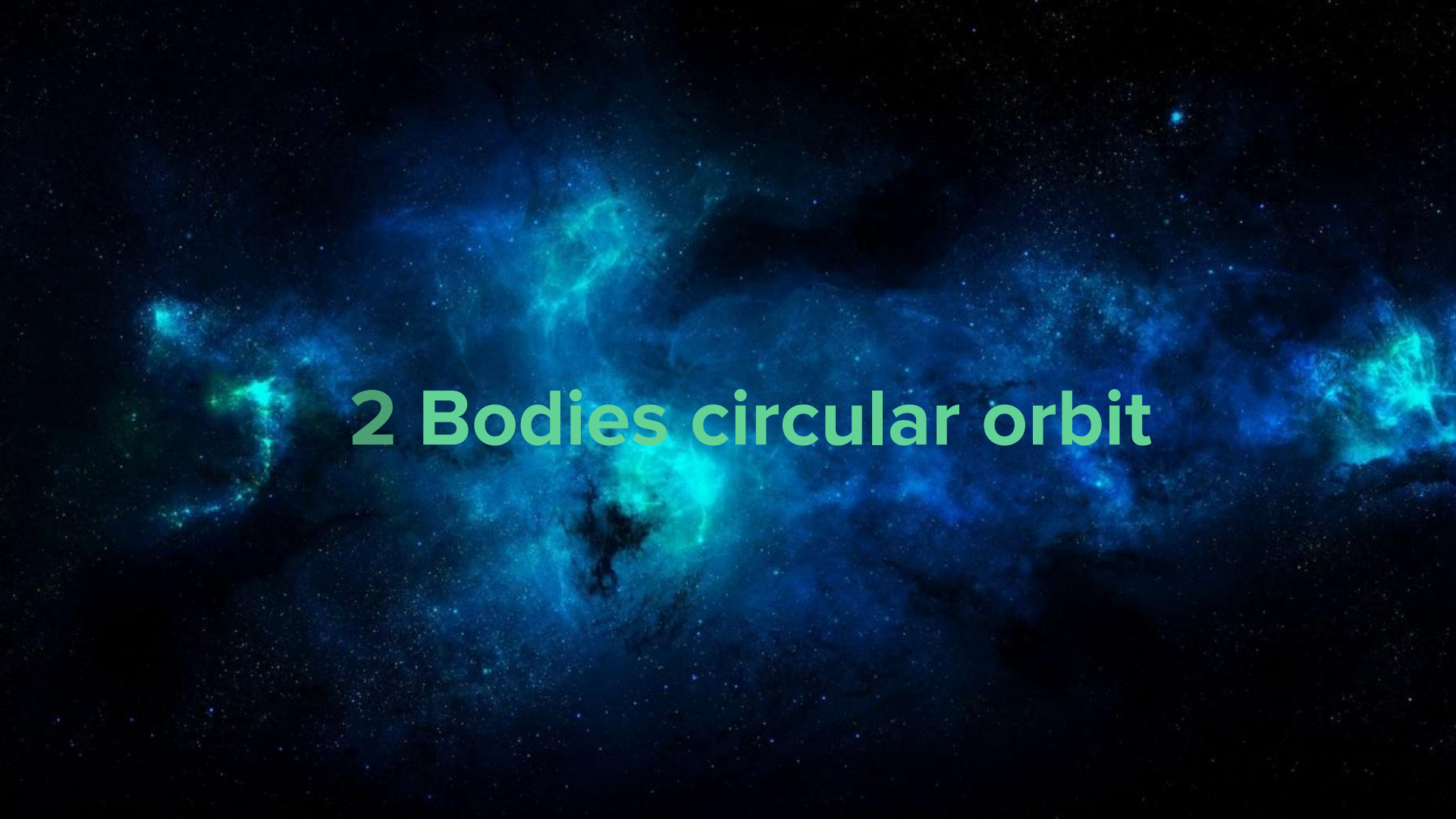


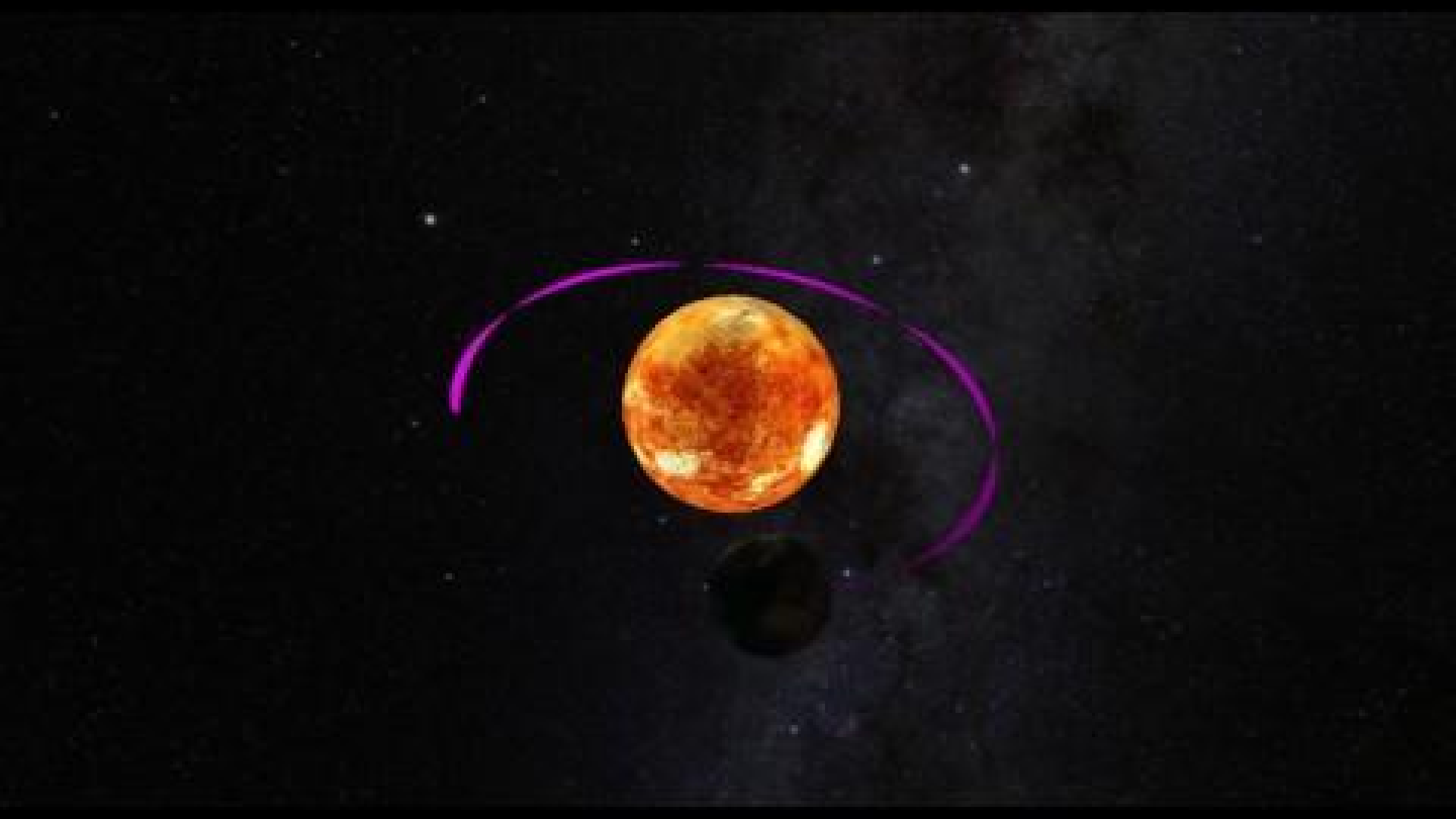


Implementation overview (RIGID BODY)

- Collisions
  - Broad phase: detect possible collisions
  - GJK: perform accurate collision detection
  - EPA: Extract the intersection vector

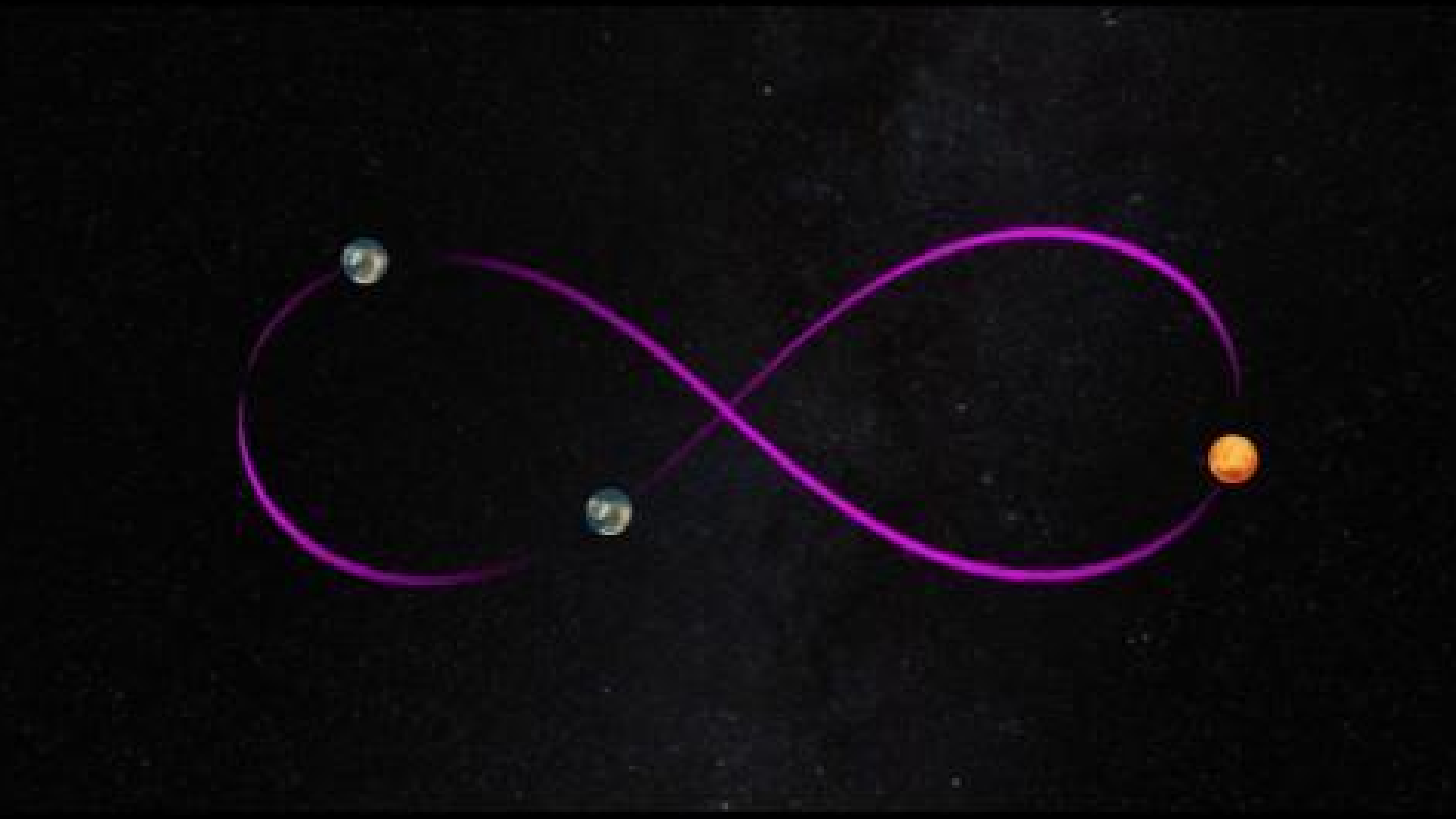



2 Bodies circular orbit

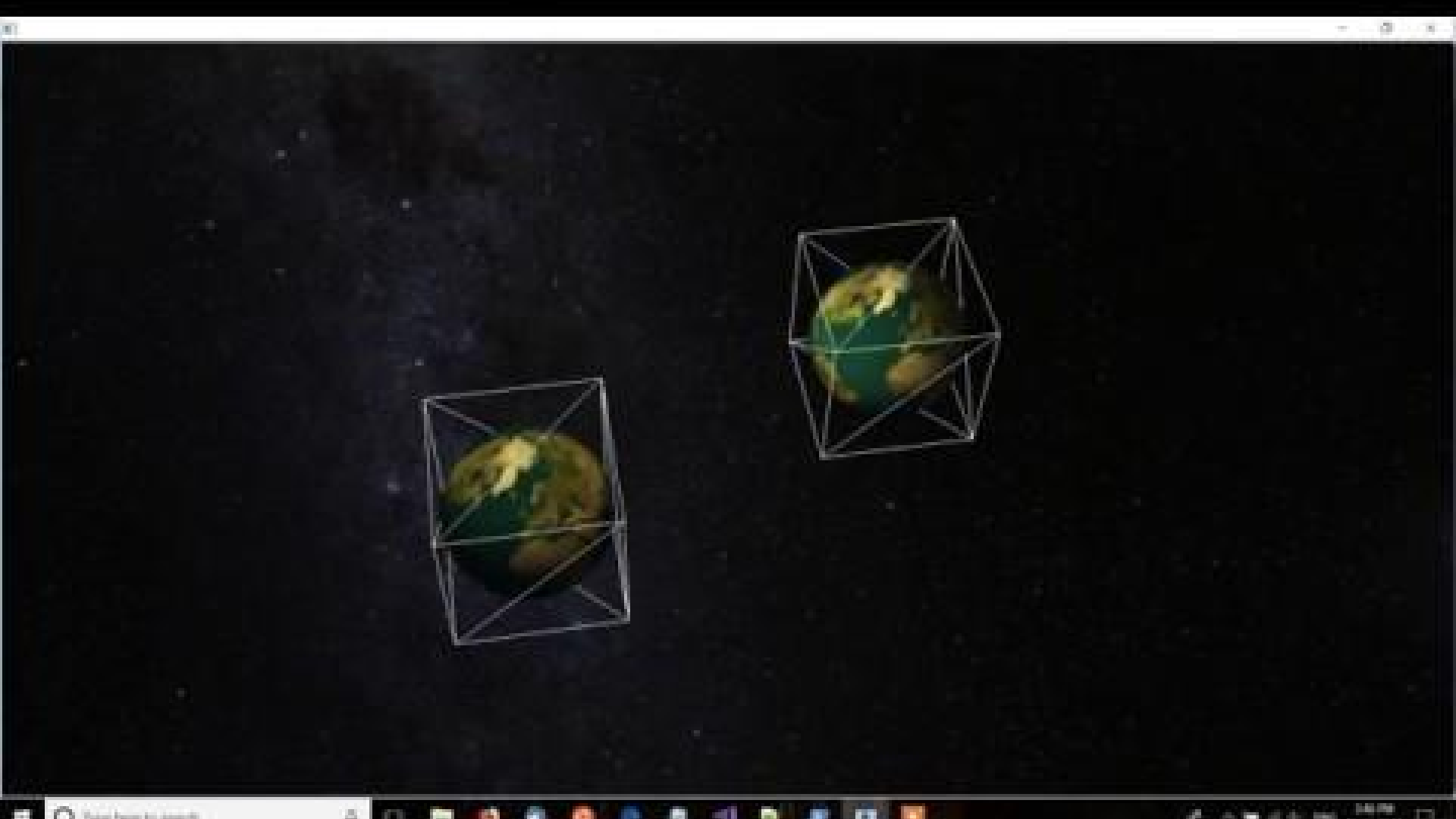


2 Bodies elliptic orbit

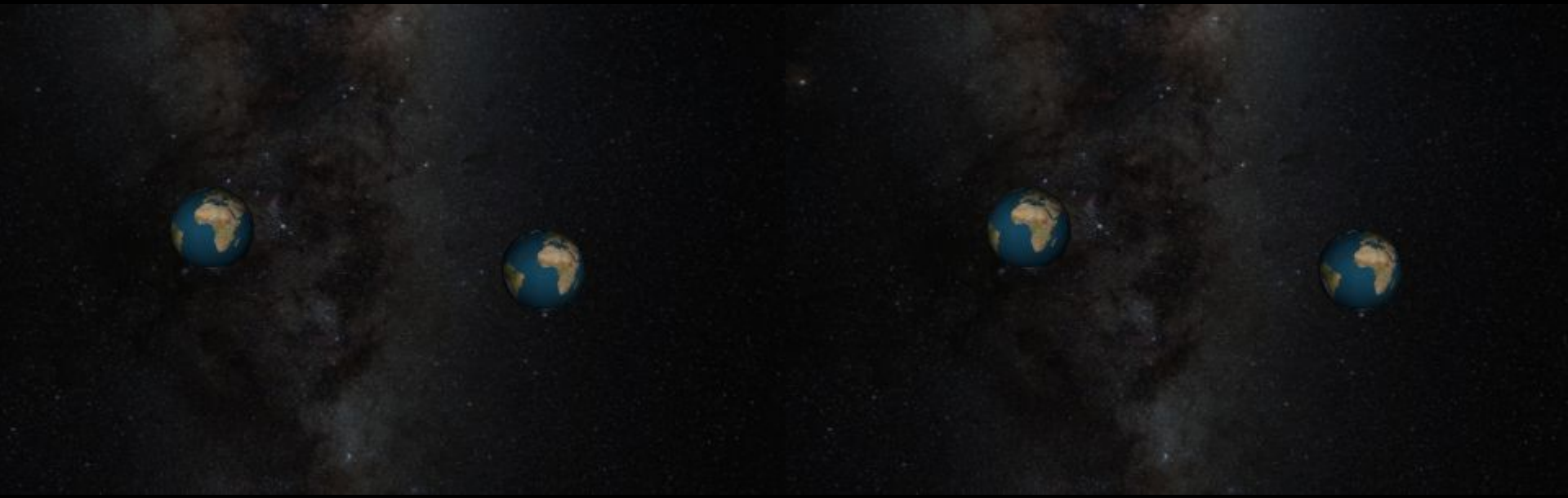
3 Bodies figure eight



Collision detection Prune and Sweep



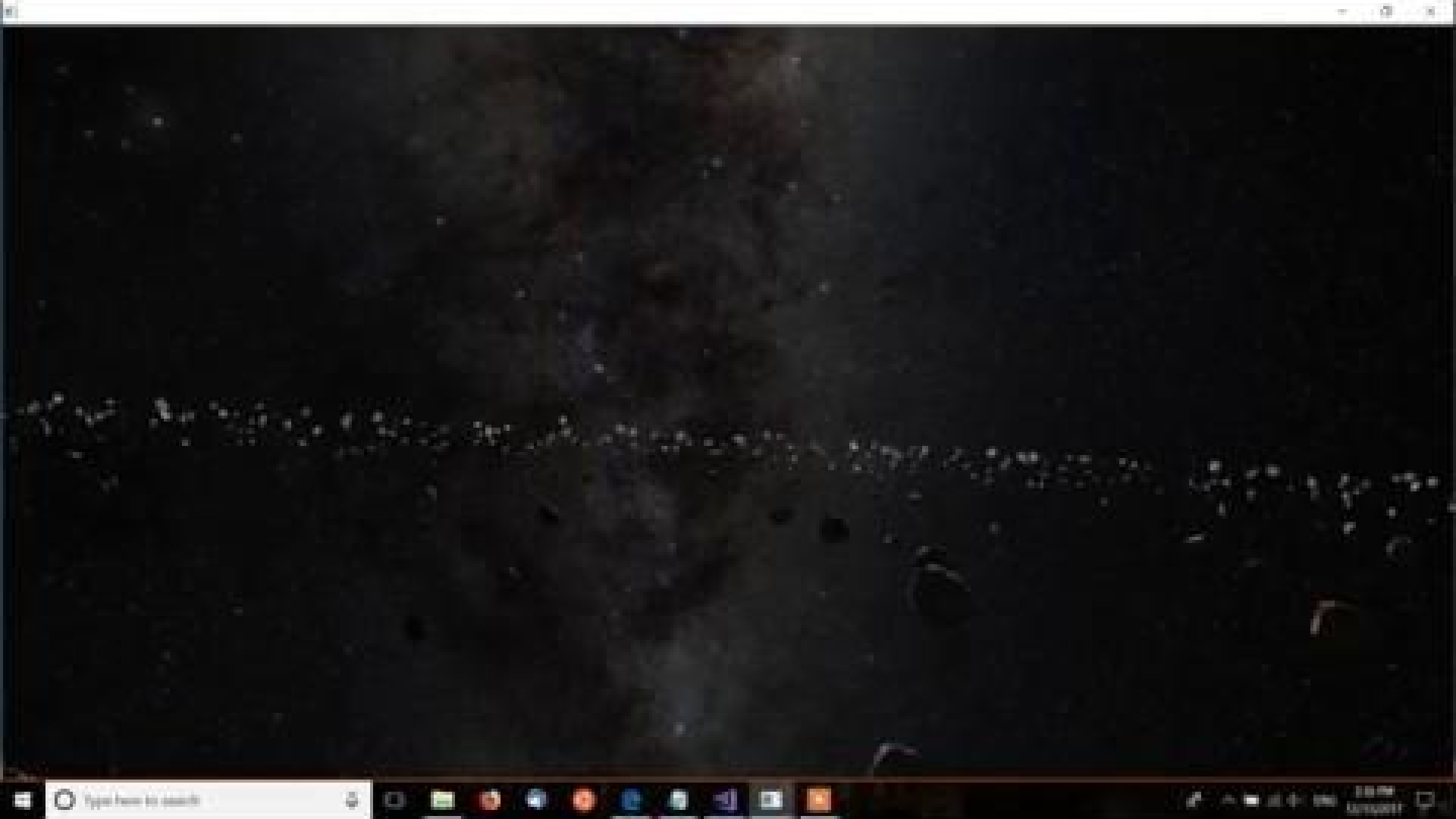
Collision detect Sphere vs. Sphere



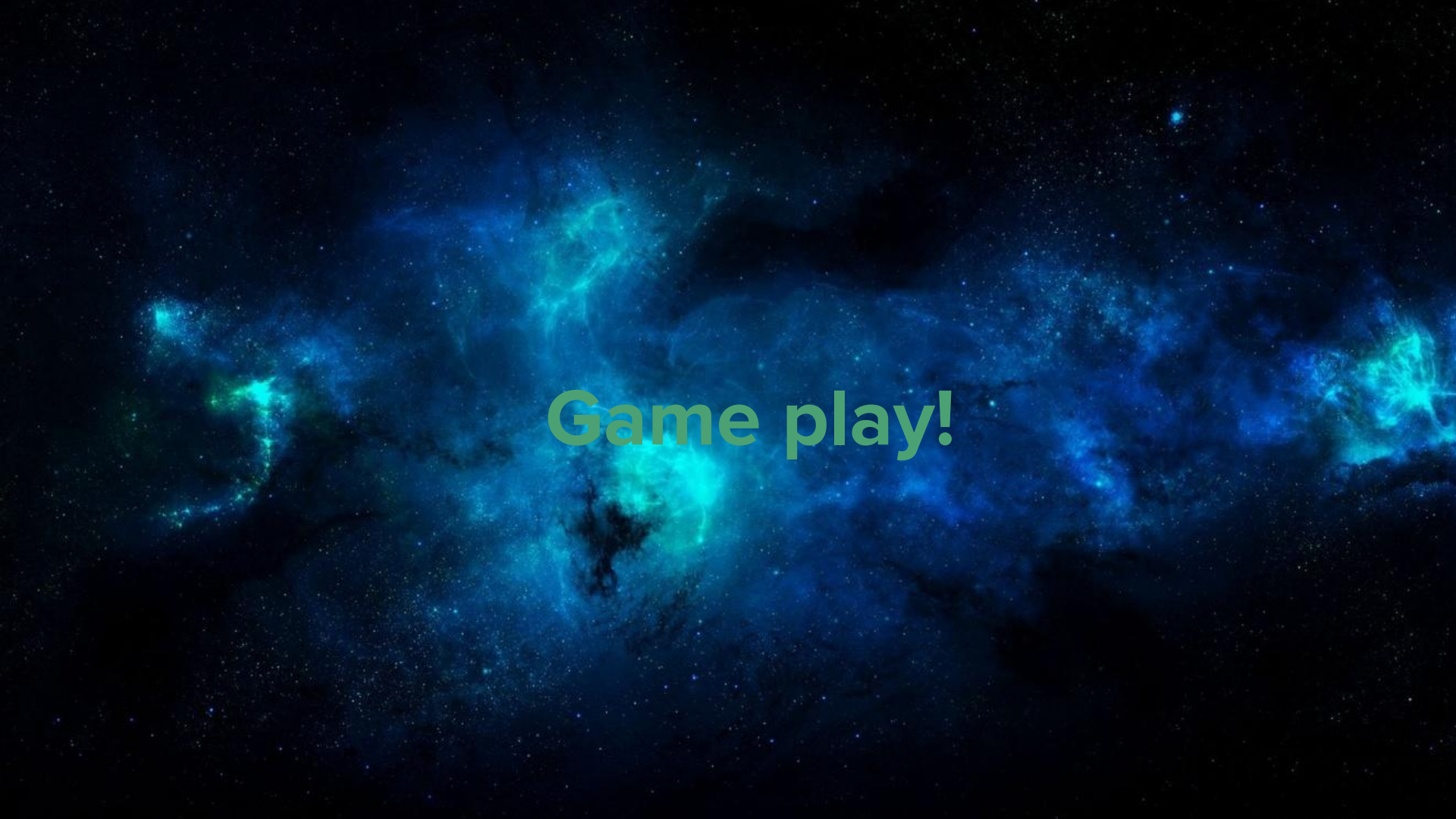
Collision detect: asteroid vs. asteroid

ADD video

Spiral pattern



Saturn


Game play!

Challenges

- Data Structure for handling N-Body systems
- Support function for GJK for not computing the Minkowski-Sum
- Apply correct collision response

Libraries

- OSG (Open Scene Graph)	Rendering
- CGAL	Geometry processing (Convex Hull)
- Boost	Start-Options
- EIGEN	Physic-calculations
- JSON for Modern C++	Scene-File parsing

Sources

- Documentations of the libraries
- Collision Detection
  - Broad Phase
    https://www.toptal.com/game/video-game-physics-part-ii-collision-detection-for-solid-objects
  - GJK (Gilbert–Johnson–Keerthi distance algorithm)
    http://programyourfaceoff.blogspot.ch/2012/01/gjk-algorithm.html
  - EPA (Expanding Polytope Algorithm)
    www.dyn4j.org/2010/05/epa-expanding-polytope-algorithm/