

# ASTEROID GAME ENGINE

Aaron de Miranda Colaço

Joston Fernandes

Kimberly Cabral

Natasha Priolkar

Prajakta Kuncolienkar

# THE ASTEROID GAME



# GAME ENGINE

- A **game engine** is a software framework designed for the creation and development of video games.
- The core functionality typically provided by a game engine includes a rendering engine ("renderer") for 2D or 3D graphics, a physics engine or collision detection (and collision response).



# Components of a game engine

- Main game program
- Rendering engine
- Audio engine
- Physics engine
- Artificial intelligence



- **Main game program:** The actual game logic has to be implemented by some algorithms. It is distinct from any rendering, sound, or input work.
- **Rendering engine:** - The rendering engine does the rendering on the display.
- **Audio engine:** The audio engine is the componentry which consists of any algorithms related to sound. It can calculate things on the CPU, or on dedicated ASIC.  
e.g. OpenAL, SDL audio, Xaudio 2, etc. are available.

- **Physics engine:** The physics engine is responsible for giving the application a realistic sense of the laws of physics in the application.
- **Artificial intelligence:** The A.I. is usually outsourced from the main game program into some special module to be designed and written by software engineers with specialist knowledge.

# WHY A GAME ENGINE?

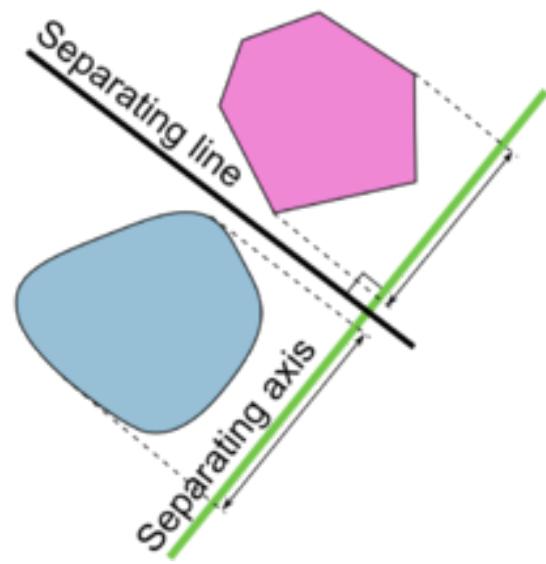
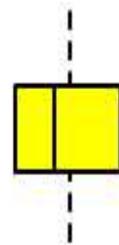
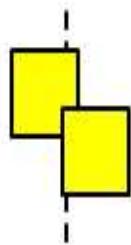
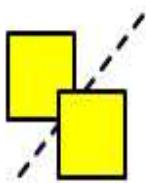
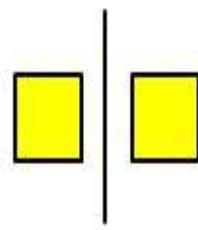
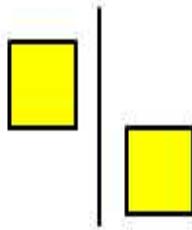
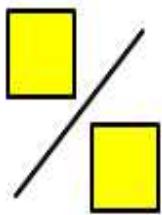


- There is a big difference between a collision engine and a physics engine. They do not do the same thing, although the physics engine generally relies on a collision engine.
- The collision engine is then split into two parts: collision detection and collision response. The latter is generally part of the physics engine.

## **Collision detection:**

- Every object has a transform and a shape (possibly multiple shapes).
- The purpose of this phase is to find out if the objects really do collide.
- Sometimes a fine level of detail is required (say, bullet collision in a shooter, where you want to be able to ignore shots that just barely miss), and also to find out exactly where the objects collide, which will affect how the objects respond.

- Algorithms used here include the Separating Axis Theorem (SAT).
- This algorithm tells us where the objects collided---how far they are penetrating each other and what the 'contact points' are.



## **Physical response :**

- At this point, a contact has been discovered, and there is enough information for the physics engine to process the contact. The physics handling can get very complex.
  
- At the most basic level, the physics engine will do something like this: it'll take the colliding objects and calculate the new positions required to separate the collided objects.



- It will move the objects to these new positions. It'll also calculate the velocity change resulting from this push, combined with restitution (bounciness) and friction values.
- The physics engine will also apply any other forces acting on the objects, such as gravity, to calculate the objects' new velocities, and their new positions.

## Contact Caching :

- With a contact cache, each set of colliding objects is saved in a lookup table.
- Each frame, when a collision is detected, this cache is queried to see if the objects were previously in contact.
- If the objects were not previously in contact, then a 'new collision' event can be generated. If the objects were previously in contact, the information can be used to provide a more stable response.

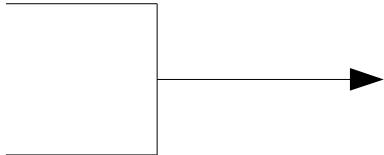


## Sleeping :

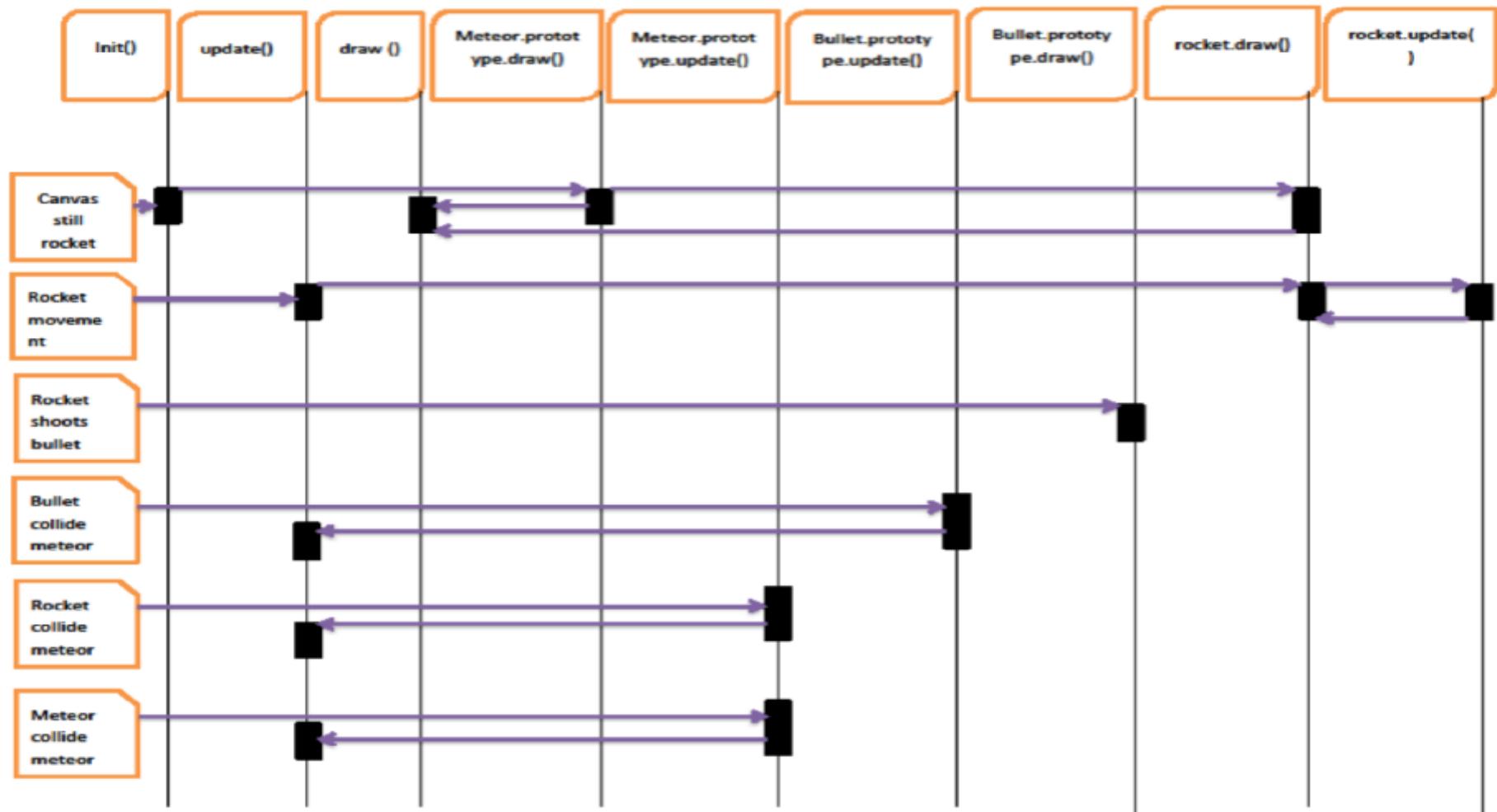
- Another very useful technique is to mark objects as being 'asleep' if they are not being interacted with.
- Sleeping objects do not get physics updates, do not collide with other sleeping objects, and basically just sit there frozen in time until another non-sleeping object collides with them.

- The impact is that all the pairs of colliding objects that are just sitting there doing nothing don't take up any processing time.
- An object is a candidate for sleeping when it has had a near-zero velocity for more than a single frame.

# Modules

- Meteor Movement
  - Rocket Movement
  - Bullet Movement
  - Collision Detection
  - Collision Response
  - Game Audio
  - Difficulty Levels
  - Lives
  - Scoreboard
- 
- Update position
  - Draw on canvas

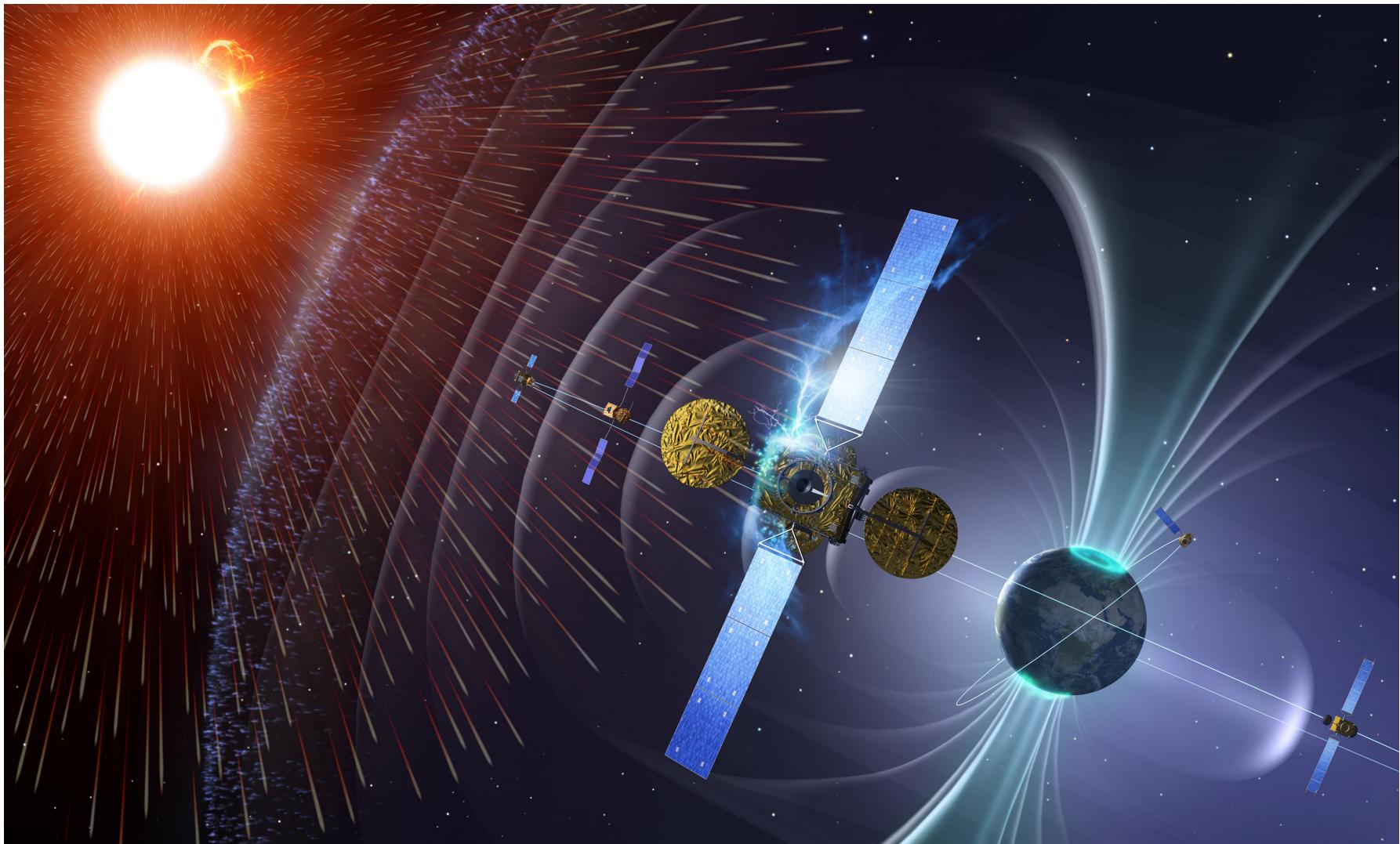
# Sequence Diagram



# THE DIFFERENCE



# Space Weather



➤ A **coronal mass ejection** (CME) is a massive burst of gas and magnetic field arising from the solar corona and being released into the solar wind.



# Solar Wind



## **Effect of space weather on space systems-**

### **Spacecraft anomalies:-**

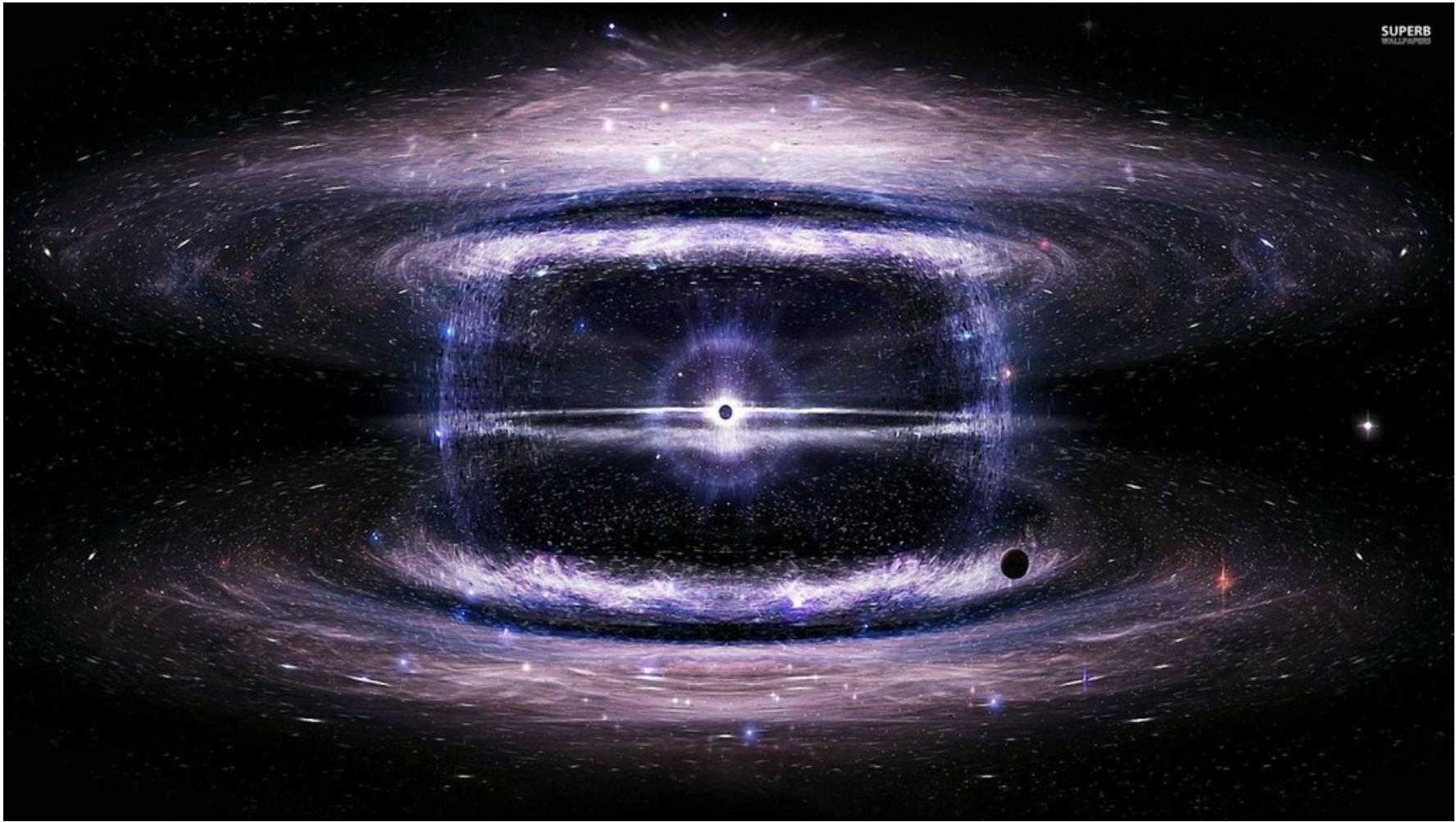
- The two most common adverse space weather effects on spacecrafts are radiation damage and spacecraft charging.
- Radiation (high energy particles) passes through the skin of the spacecraft and into the electronic components.

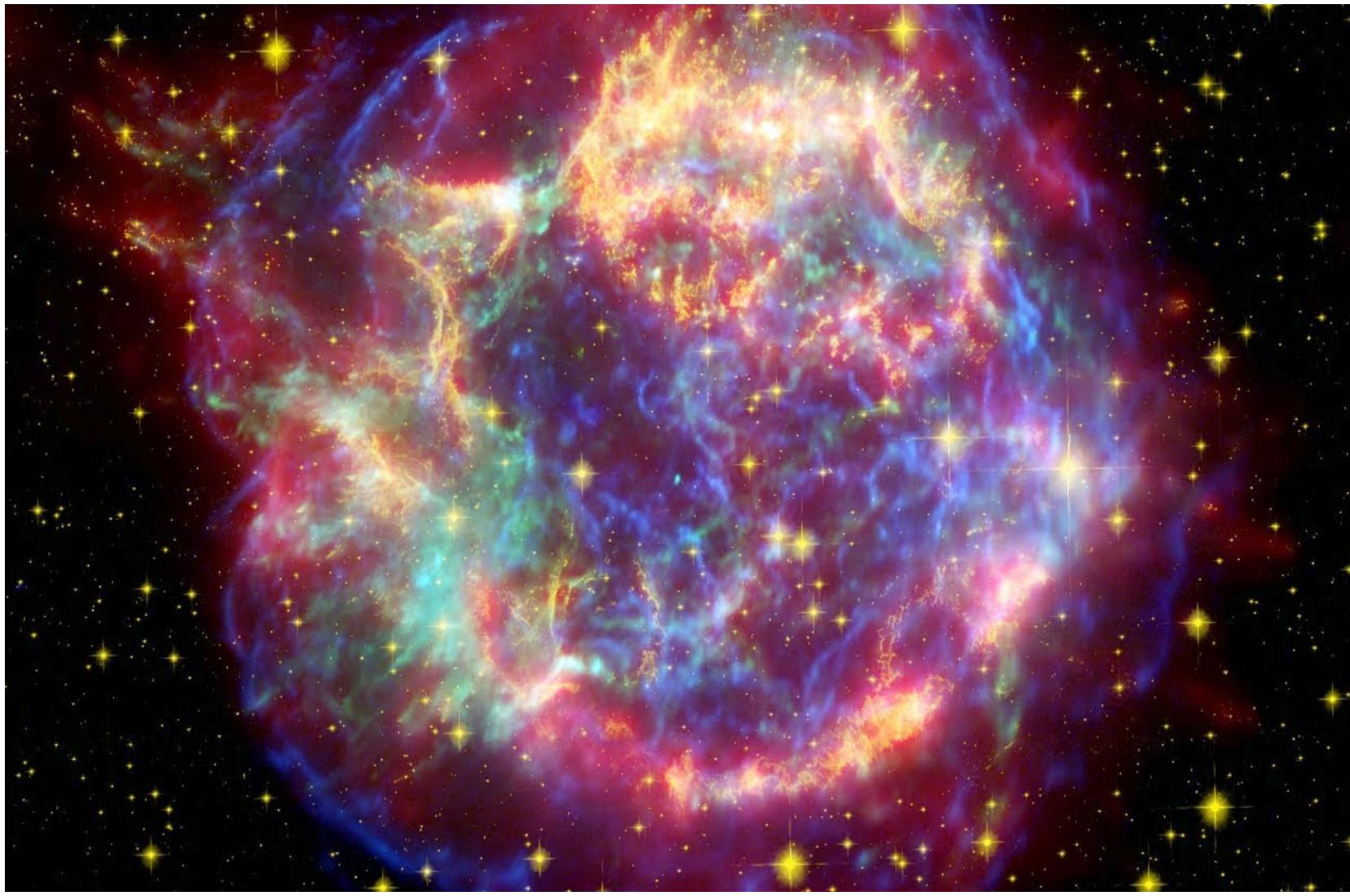




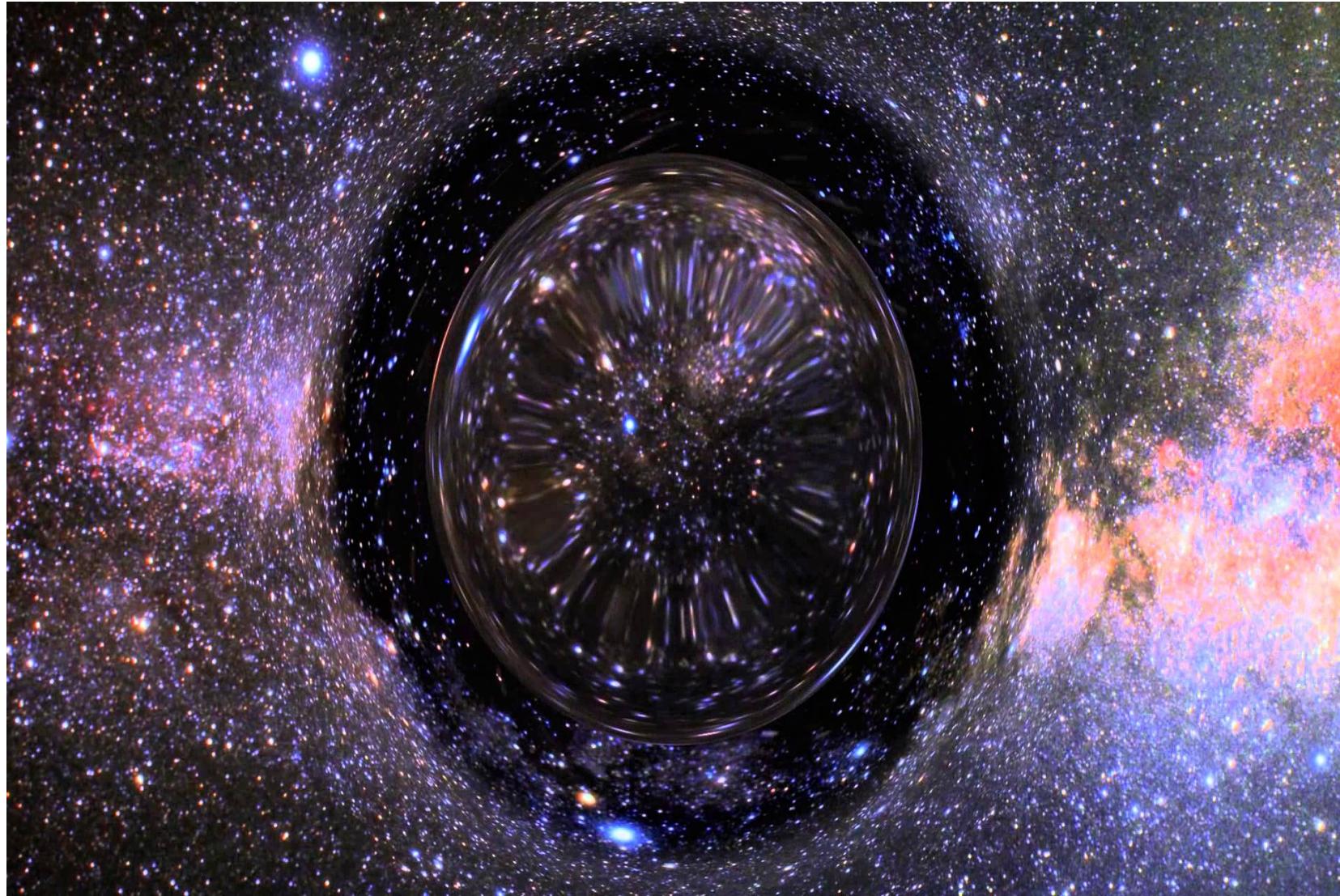
- Spacecraft charging is the accumulation of an electrostatic charge on a non-conducting material on the spacecraft's surface by low energy particles.
- If enough charge is built-up, a discharge (spark) occurs.

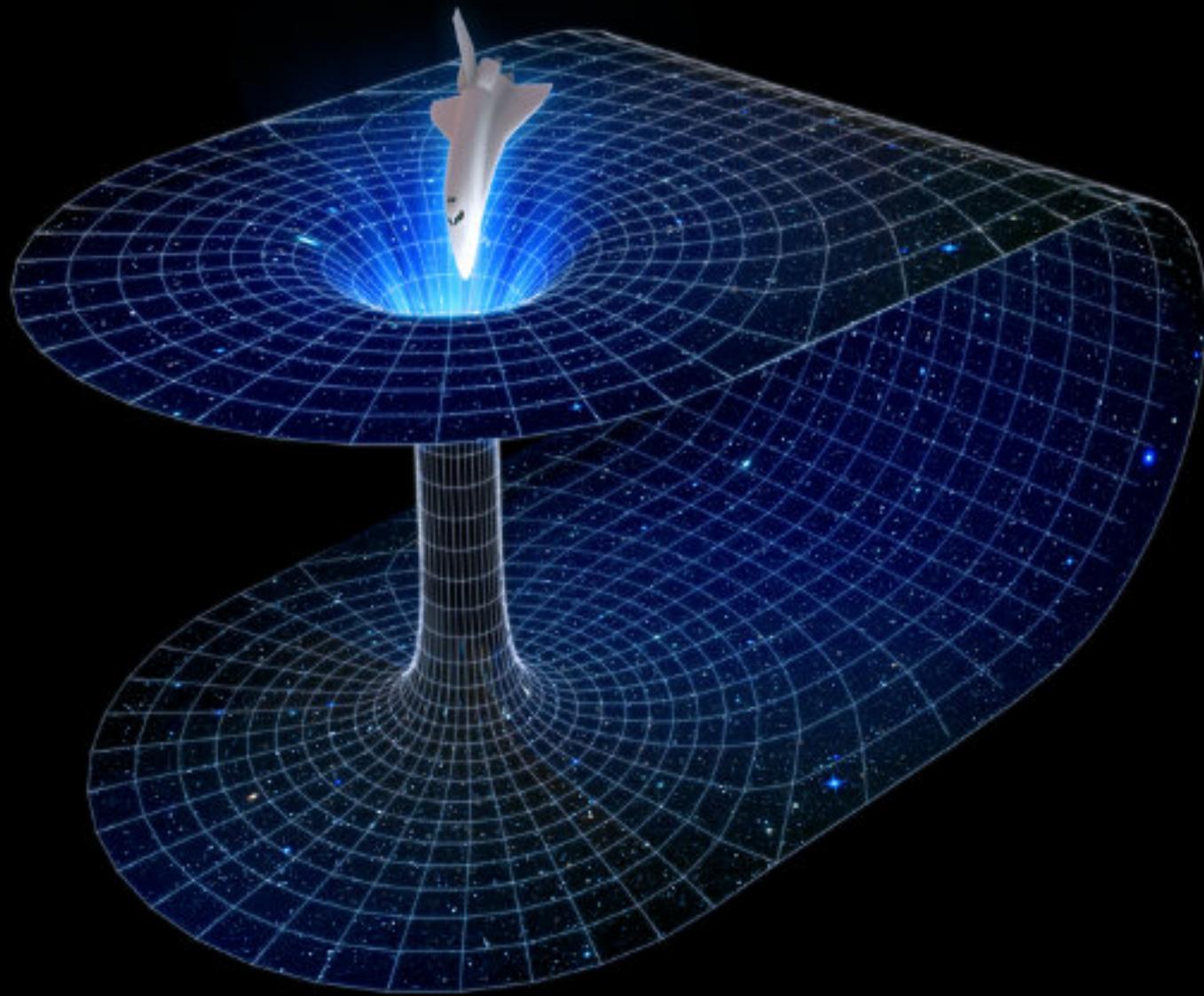
# Supernova





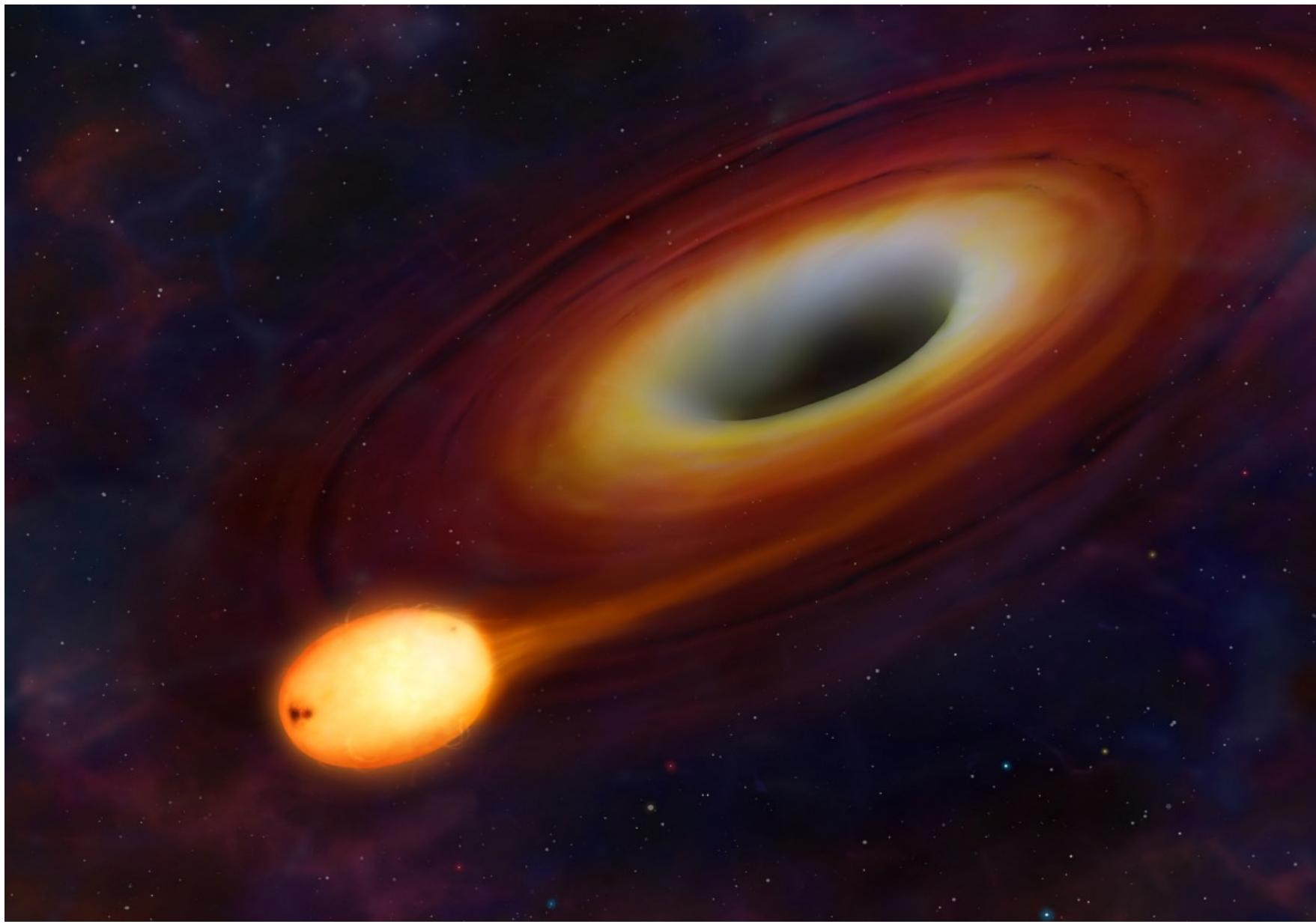
# Wormhole





# Blackhole





**THANK YOU!**