

# NEXUS

Proyecto práctico en Python aplicado a la  
Astrodinámica Aeroespacial

Karol Villafrades Santos

Luisa María Franco Gómez

Programa de Ingeniería Aeroespacial

Universidad de Antioquia

Facultad de Ingeniería — Seccional Oriente

REINGRESO AEROESPACIAL

Diciembre 2 de 2025



# Introducción:

- **NEXUS** es un videojuego desarrollado en Python usando Pygame.
- Simula el reingreso aeroespacial de una nave hacia distintos planetas.
- Incluye física, partículas, colisiones, HUD, meteoritos y combustible.
- El objetivo: completar los 8 planetas del sistema solar.

**NEXUS**  
REINGRESO AEROESPACIAL

# Funcionamiento General:



- El jugador controla un vehículo aeroespacial.
- Debe aterrizar suavemente en cada planeta.
- Evitar:
  - Meteoritos.
  - Falta de combustible.
  - Aterrizajes con demasiada velocidad.
- Cada planeta tiene diseño, colores y rotación propia.

# Estados del Juego:

- MENÚ
- JUGANDO
- PAUSA
- NIVEL COMPLETADO
- CARGANDO NIVEL
- TERMINADO
- GAME OVER



NEXUS

REINGRESO AEREOESPACIAL

# Mecánicas Principales:

- Física realista: gravedad, impulso y frenado.
- Combustible limitado.
- Rotación planetaria optimizada con caché.
- Meteoritos y estrellas de combustible.
- Sistema de vidas y puntaje.
- Explosiones con partículas.



# Entradas y Salidas:

Entradas	Salidas
<ul style="list-style-type: none"><li>• Flechas: movimiento.</li><li>• P: pausa.</li><li>• Mouse: botones de interfaz.</li></ul>	<ul style="list-style-type: none"><li>• Animaciones.</li><li>• HUD dinámico.</li><li>• Pantallas de nivel y Game Over.</li></ul>

NEXUS  
REINGRESO AEREOESPACIAL

# Retos Encontrados:



- Mantener múltiples estados sin conflictos.
- Reiniciar niveles sin regresar siempre al nivel 1.
- Optimizar rotación del planeta (alto costo en CPU).
- Controlar crecimiento de partículas/meteoritos.
- Colisiones precisas con planetas/meteoritos.
- Manejar combustible sin fallos.

# Soluciones Implementadas:

- Máquina de estados clara y estable.
- Instrucciones al iniciar el juego.
- Caché de rotación del planeta con ROTATE EVERY.
- Límites: MAX PARTICULAS, MAX METEORITOS.
- Criterio de aterrizaje: velocidades absolutas.
- Botones funcionales y menú interactivo.



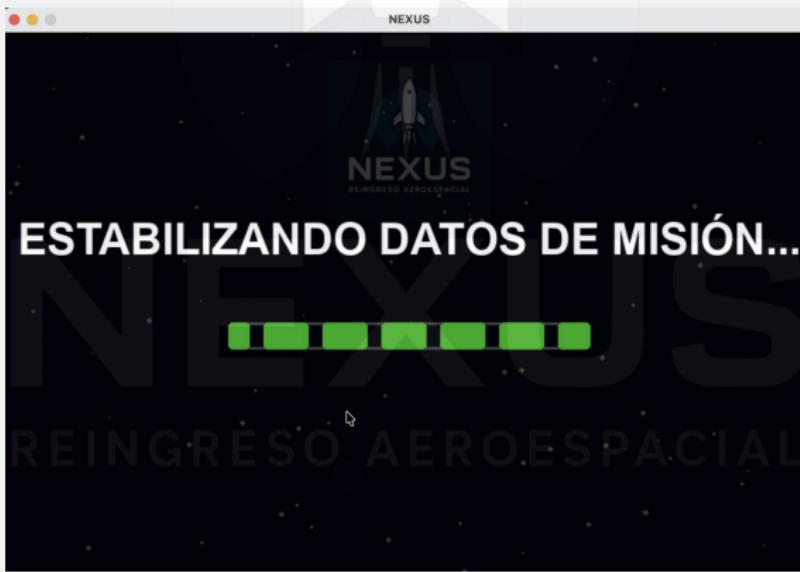
# Aprendizajes:



- Uso de librerías gráficas (Pygame).
- Programación orientada a eventos.
- Optimización de renderizado.
- Matemáticas aplicadas a física simple.
- Diseño modular y estructuras limpias.

# Investigación Independiente:

- Efectos de partículas personalizados.
- Técnicas de rotación optimizada.
- Diseño visual de planetas y nebulosas.
- Flujo de pantallas estilo videojuego profesional.



# Librerías Utilizadas:

- **pygame**: motor multimedia.
- **random**: generación de valores aleatorios.
- **math**: funciones matemáticas.
- **sys**: control del sistema.



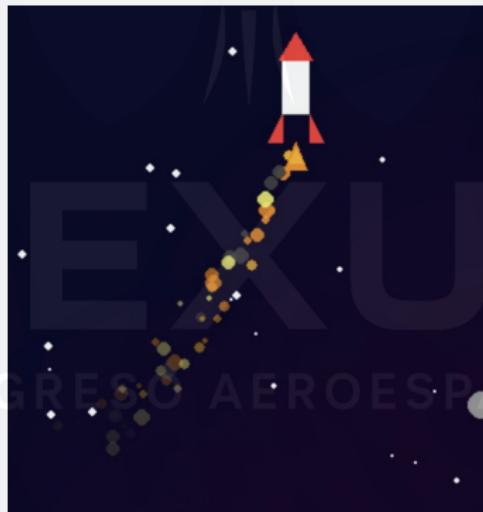
# Estructura del Código:



- Variables globales del juego.
- Definición de planetas y colores.
- Funciones:
  - dibujar planeta.
  - reiniciar juego.
  - dibujar barra superior.
  - manejar click botones.
- Bucle principal del juego.

# Muestra de Funcionamiento:

- Nave descendiendo con gravedad.
- Meteoritos cayendo según nivel.
- Estrellas de combustible.
- Explosiones con partículas.
- Pantallas de transición y nivel completado.



# Documentación y Entrega

- Código: `nexus.py`
- Archivos: `logo.png`, `udea_logo.png`
- `requirements.txt`
- `README.md`

NEXUS  
REINGRESO AEREOESPACIAL

# Conclusión

- NEXUS es un videojuego completo y funcional.
- Integra física, UI, estados y animaciones avanzadas.
- Representa un logro técnico sólido en Pygame.

NEXUS  
REINGRESO AEREOESPACIAL