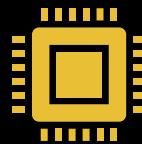


# DREAM-TO-WORLD ENGINE



Transforming Imagination  
into Interactive Reality



- Design an intelligent creative engine that converts natural language prompts into immersive 3D worlds.

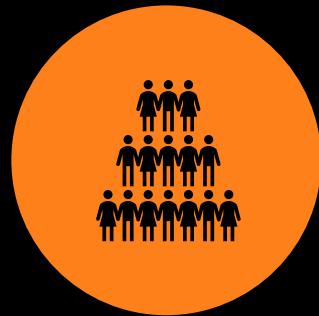


- Integrate AI-assisted design tools with interactive simulation and real-time rendering.



- Develop a modular, scalable system that supports user-generated virtual environments.

# TEAM AND CONTACT INFO



TEAM MEMBER: ZYAD  
ALKURDI



EMAIL:  
[ALKURDZM@MAIL.UC.EDU](mailto:ALKURDZM@MAIL.UC.EDU)



PROJECT ADVISOR: HRISHI  
BHIDE

# PROJECT ABSTRACT



The Dream-to-World Engine is an intelligent system that generates interactive 3D worlds directly from text-based input.



By combining natural language processing, procedural generation, and real-time rendering, the project allows users to describe imagined settings and instantly visualize them as explorable environments.



The system emphasizes modular design, scalability, and creative freedom.

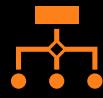


It supports dynamic lighting, environmental effects, and object interaction, making it adaptable for both entertainment and educational use cases.



The goal is to bridge human imagination and digital creation through seamless, intuitive world generation.

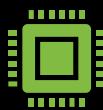
# USER STORIES



- As a user, I want to input a written scene description so I can see it generated as a 3D environment.



- As a designer, I want modular control over terrain, lighting, and models to fine-tune results.



- As a developer, I want reusable APIs to connect AI text interpretation to 3D object libraries.



- As a viewer, I want to interact with the generated environment to explore freely.

# DESIGN DIAGRAM



Core Components:



- Input Processor: parses natural language into structured scene data.



- Scene Builder: maps parsed data to 3D assets and spatial layout.



- Renderer: uses Three.js for dynamic visualization.



- Interaction Layer: supports camera movement and user interaction.

# MAJOR PROJECT CONSTRAINTS

- Economic: Limited personal budget; relying on free/open-source software like Three.js and Blender assets.

- Technical: Requires efficient GPU rendering and NLP integration within browser limitations.

- Ethical: Must ensure generated content avoids copyrighted or unsafe materials.

- Time: Balancing academic deadlines and development milestones for full functionality by Expo.

# PROJECT PROGRESS



Current State:



- Base Three.js rendering engine implemented.



- Prototype sun and astronaut models completed with texture and lighting effects.

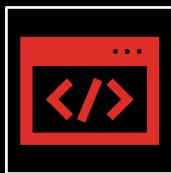


- Interactive navigation and camera animation features functional.



- Working hover and click event handling for object interaction.

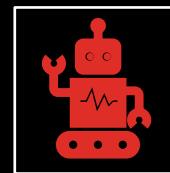
# EXPECTED ACCOMPLISHMENTS



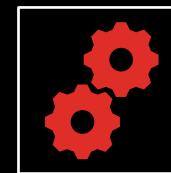
- Complete modular world generation from text prompts.



- Add terrain, weather, and lighting variation features.



- Integrate AI model for semantic parsing and object placement.



- Optimize performance and implement user interface polish for demo.

# DIVISION OF WORK



SINCE SOLO  
PROJECT — SELF-  
ASSIGNED ROLES:



- FRONTEND  
(THREE.JS, UI,  
INTERACTION): ZYAD  
ALKURDI



- BACKEND  
(PARSING, DATA  
STRUCTURES): ZYAD  
ALKURDI



- TESTING AND  
OPTIMIZATION: ZYAD  
ALKURDI



- DOCUMENTATION  
AND PRESENTATION:  
ZYAD ALKURDI

# EXPECTED EXPO DEMO



- Live text-to-3D generation walkthrough.



- User enters a short scene description (e.g., "a sunset beach with a lighthouse").



- Engine constructs and renders a 3D scene in real time.



- Demonstration of camera animation and object interaction.



- Q&A on technical design and AI integration plans.