🦇 Dungeon Ecosystem 3D Engine

A scientifically-accurate, real-time 3D simulation of dungeon ecosystems with complete predator-prey dynamics, environmental modeling, and procedural generation.

Show Image

Show Image

Show Image

🎯 Vision

This project creates a living, breathing dungeon ecosystem where:

- **Species evolve** based on environmental pressures and genetic drift
- **Populations fluctuate** realistically through predator-prey cycles
- **Environment matters** - humidity, temperature, and resources drive creature behavior
- **Migration happens** when populations exceed carrying capacity
- **Everything is connected** - removing one species cascades through the entire ecosystem

Unlike traditional game ecosystems that use simple spawning rules, this simulation models actual ecological relationships with mathematical accuracy.

🚀 Quick Start

Prerequisites

- Node.js 16+
- Modern browser with WebGL2 support

- Basic understanding of JavaScript ES6 modules

Installation

```bash
git clone https://github.com/your-org/dungeon-ecosystem-3d
cd dungeon-ecosystem-3d
npm install
npm run dev
```

First Run

1. Open `http://localhost:3000` in your browser
2. You should see a procedurally generated dungeon
3. Watch populations change in real-time
4. Click rooms to inspect individual ecosystems

📋 Current Status

✅ Completed (Phase 1)

- **Core Math Library** - Vector3, Matrix4, Quaternion, MathUtils with full test coverage
- **Ecosystem Mathematics** - Population dynamics, predation rates, environmental gradients
- **Project Architecture** - Complete folder structure and development roadmap

🔧 In Development (Phase 2)

- WebGL2 rendering pipeline
- Basic shader system
- Mesh generation and management

📅 Upcoming (Phases 3-7)

- Procedural dungeon generation
- Multi-threaded ecosystem simulation
- Creature AI and flocking behavior
- Real-time 3D visualization
- Interactive ecosystem manipulation

See ROADMAP.md for detailed development timeline.

🏗️ Architecture Overview

```
dungeon-ecosystem-3d/
├── src/
│   ├── math/          # Core mathematics (✅ Complete)
│   ├── rendering/     # 3D graphics pipeline (🔧 In Progress)
│   ├── simulation/    # Ecosystem simulation (📅 Planned)
│   ├── entities/      # Creatures and objects (📅 Planned)
│   ├── generation/    # Procedural content (📅 Planned)
│   └── workers/       # Background simulation (📅 Planned)
├── assets/            # Shaders, models, textures
├── tools/             # Development utilities
├── tests/             # Comprehensive test suite
└── docs/              # Documentation
```

Key Design Principles

1. **Simulation-First**: The ecosystem simulation is completely separate from rendering, allowing for headless testing and different visualization modes.
2. **Web Worker Architecture**: Heavy calculations run in background threads to maintain smooth 60fps visualization.
3. **Data-Driven**: Species behaviors, environmental parameters, and generation rules are defined in JSON files for easy experimentation.
4. **Scientific Accuracy**: Based on real ecological models from population biology, environmental science, and evolutionary theory.

🧬 Ecosystem Model

Species Hierarchy

- **Primary Producers**: Cave moss, slimes (feed on organic matter)
- **Primary Consumers**: Beetles, small insects (eat producers)
- **Secondary Consumers**: Spiders, rats (eat primary consumers)
- **Decomposers**: Bacteria, specialized organisms (recycle nutrients)

Environmental Factors

- **Temperature**: Affects metabolism and creature activity
- **Humidity**: Essential for moss growth and creature survival
- **Air Flow**: Distributes scents, affects gas concentrations

- **Water Sources**: Create humidity gradients, support specific species
- **Organic Matter**: Food source for decomposers and producers
- **Light Penetration**: Affects photosynthetic organisms

Population Dynamics

```javascript
// Logistic growth with environmental carrying capacity
dN/dt = rN(1 - N/K) - predation - migration

// Predation follows Lotka-Volterra dynamics
predationRate = efficiency × predators × prey / (prey + 1)

// Environmental suitability affects carrying capacity
K = baseCapacity × environmentalSuitability
```

🔬 Running Tests

```bash
# Run all tests
npm test

# Run specific test suites
npm run test:math
npm run test:simulation
npm run test:rendering

# Run with coverage
npm run test:coverage

# Continuous testing during development
npm run test:watch
```

Test Structure

- **Unit Tests**: Individual class functionality
- **Integration Tests**: Cross-system compatibility
- **Ecosystem Tests**: Simulation stability and realism
- **Performance Tests**: Frame rate and computation benchmarks

🛠️ Development Setup

Required Tools

- **VS Code** (recommended) with extensions:
  - ES6 String HTML for shader syntax highlighting
  - WebGL GLSL Editor for shader development
  - Live Server for local development

Development Workflow

1. **Morning**: Implement current roadmap step
2. **Midday**: Write/run tests for new functionality
3. **Afternoon**: Update documentation and commit changes
4. **Evening**: Plan next day's work and update project status

Code Style

- ES6 modules throughout
- JSDoc comments for all public methods
- Consistent naming: `camelCase` for variables, `PascalCase` for classes
- Performance-critical code includes in-place operation variants
- All magical numbers defined as named constants

📊 Performance Targets

- **Rendering**: Smooth 60fps with 1000+ creatures visible
- **Simulation**: Handle 10,000+ individual organisms across 50+ rooms
- **Startup**: Load and initialize complete ecosystem in <3 seconds
- **Memory**: Stay under 500MB for typical dungeon complexity

🤝 Contributing

Getting Started

1. Read this README thoroughly
2. Study the Architecture Guide
3. Review the Development Roadmap
4. Look at existing tests to understand code patterns

5. Start with small improvements or bug fixes

Contribution Process

1. **Fork** the repository
2. **Create branch** with descriptive name (`feature/creature-ai` or `fix/population-crash`)
3. **Implement changes** following existing code patterns
4. **Add tests** for new functionality
5. **Update documentation** as needed
6. **Submit pull request** with clear description

Areas Needing Help

- ☐ 3D model creation for creatures and environments
- ☐ Shader optimization for large populations
- ☐ Advanced AI behaviors (territorial, mating, learning)
- ☐ Sound design and spatial audio
- ☐ Mobile device optimization
- ☐ Advanced ecological models (disease, parasitism, mutualism)

📚 Learning Resources

Ecosystem Simulation

- Population Biology Primer
- Environmental Modeling Guide
- Species Interaction Types

3D Graphics

- WebGL2 Reference
- Shader Development
- 3D Math Explained

Game Development

- Entity-Component Systems
- Performance Optimization
- Procedural Generation

🐛 Known Issues

- Math library is complete but WebGL context not yet implemented
- Procedural generation algorithms still in design phase
- No mobile device testing yet
- Audio system not designed

See ISSUES.md for complete list and workarounds.

📈 Roadmap Summary

Phase	Focus	Duration	Status
1	Foundation (Math, Setup)	2 weeks	✅ 50% Complete
2	Rendering Pipeline	2 weeks	🔧 In Progress
3	Procedural Generation	2 weeks	📅 Planned
4	Core Simulation	3 weeks	📅 Planned
5	Creature Visualization	2 weeks	📅 Planned
6	Advanced Features	2 weeks	📅 Planned
7	Polish & Optimization	3 weeks	📅 Planned

Total Development Time: ~16 weeks for full feature completion

🎮 Demo Features (When Complete)

- **God Mode**: Fly through dungeons, observe ecosystem from above
- **Species Tracker**: Follow individual creatures through their lifecycle
- **Time Controls**: Speed up/slow down/rewind ecosystem development
- **Environmental Tools**: Modify temperature, humidity, add/remove resources
- **Population Graphs**: Real-time visualization of population dynamics
- **Migration Maps**: Watch species spread through dungeon networks
- **Evolutionary Trees**: See how species adapt over generations

📞 Support & Community

- **Primary Developer**: [Your contact info]
- **Project Discussions**: [GitHub Discussions link]
- **Bug Reports**: [GitHub Issues link]

- **Development Blog**: [Blog/devlog link]
- **Discord Community**: [Discord invite]

📄 License

This project is licensed under the MIT License - see LICENSE for details.

🙏 Acknowledgments

- **Ecological Models**: Based on research from population biology and systems ecology
- **3D Graphics**: Inspired by modern game engine architecture
- **Community**: Thanks to early contributors and testers
- **Research**: Special thanks to academic papers that informed our ecosystem models

---

"*Creating digital life that behaves like real life - one algorithm at a time.*"

Last Updated: Phase 1 Progress - Core math library complete, WebGL pipeline in development