# Lyra - Interactive 3D Virtual Companion

A stunning React-based web application featuring an interactive 3D virtual companion with speech synthesis, character animations, and a beautiful cosmic-themed user interface.

## 🌟 Features

### 3D Character Scene

* **Interactive 3D Environment**: Built with react-three-fiber and drei
* **Character Model**: Custom 3D character with idle animations and expressions
* **Real-time Animations**: Breathing, blinking, wave gestures, and emotional expressions
* **Mouse Interaction**: Click to interact with the character
* **Responsive 3D Camera**: Subtle camera movements based on mouse position

### Voice System

* **Text-to-Speech (TTS)**: Web Speech API integration with multiple voice options
* **Lip-Sync Visualization**: Real-time audio analysis for mouth movements
* **Voice Customization**: Adjustable rate, pitch, and volume controls
* **Multiple Voice Support**: Access to system voices with language selection
* **Error Handling**: Graceful fallback with visual notifications when TTS fails

### Interaction System

* **Smart Detection**: Mouse clicks, keyboard events, and idle time tracking
* **Contextual Responses**: Different dialogue based on user actions
* **Personality System**: Character responds with humor and personality
* **Quick Actions**: Pre-defined interaction buttons for common actions
* **Idle Behavior**: Character responds to inactivity with appropriate dialogue

### Cosmic UI Theme

* **Dark Space Theme**: Deep blue and purple cosmic backgrounds
* **Star Field Animation**: Dynamic star field with twinkling effects
* **Constellation Patterns**: Animated constellation lines and cosmic dust
* **Nebula Effects**: Gradient backgrounds with cosmic atmosphere
* **Glowing UI Elements**: Cyan and purple accent colors with glow effects
* **Responsive Design**: Mobile-friendly interface with adaptive layouts

### Character Expressions & Animations

* **Facial Expressions**: Happy, surprised, sad, and neutral expressions
* **Body Animations**: Wave gestures, idle movements, and breathing
* **Emotion System**: Character reacts with appropriate expressions
* **Animation Blending**: Smooth transitions between different states

### State Management

* **Zustand Store**: Efficient state management for character, voice, and UI
* **Real-time Updates**: Reactive updates across all components
* **Persistent Settings**: Character and voice preferences maintained

## 🚀 Live Demo

**Deployed Application**: <https://pbnp8zcym8.space.minimax.io>

## 🛠️ Technology Stack

### Core Technologies

* **React 18.3** - Modern React with hooks and functional components
* **TypeScript** - Type-safe development
* **Vite** - Fast build tool and development server
* **Tailwind CSS** - Utility-first CSS framework

### 3D Graphics

* **Three.js 0.160.0** - 3D graphics library
* **react-three-fiber 8.15.19** - React renderer for Three.js
* **react-three/drei 9.88.13** - Useful helpers and components
* **@pixiv/three-vrm** - VRM model support

### Animation & Interaction

* **Framer Motion** - Animation library for React
* **react-spring** - Spring-physics animations
* **Zustand** - Lightweight state management

### Audio & Speech

* **Web Speech API** - Browser-native text-to-speech
* **Web Audio API** - Audio analysis for lip-sync

### UI Components

* **Lucide React** - Beautiful icon library
* **react-error-boundary** - Error handling boundaries

## 📁 Project Structure

/src  
├── components/  
│ ├── Character/  
│ │ ├── Character.tsx # Main 3D character component  
│ │ ├── CharacterScene.tsx # 3D scene setup and controls  
│ │ └── CosmicEnvironment.tsx # Space background and effects  
│ └── UI/  
│ ├── MainInterface.tsx # Main UI overlay and controls  
│ └── CosmicBackground.tsx # Animated background elements  
├── hooks/  
│ └── useInteractions.ts # User interaction detection and handling  
├── lib/  
│ └── tts.ts # Text-to-speech and dialogue management  
├── store/  
│ └── useStore.ts # Zustand state management  
├── App.tsx # Main application component  
└── main.tsx # Application entry point

## 🎮 How to Use

### Basic Interaction

1. **Click on Lyra**: Direct interaction with the character
2. **Use Quick Actions**: Bottom action bar for common interactions
3. **Open Menu**: Top-right menu button for settings
4. **Idle Detection**: Lyra responds when you’re inactive

### Quick Action Buttons

* **💬 Greet**: Friendly greeting from Lyra
* **😄 Joke**: Request a humorous response
* **❤️ Compliment**: Receive a nice compliment
* **🕐 Time**: Ask for the current time

### Character Controls

* **Expressions**: Happy, surprised, sad, neutral
* **Animations**: Wave, idle, breathing
* **Voice Settings**: Rate, pitch, volume adjustment

## 🔧 Development

### Prerequisites

* Node.js 18+
* pnpm (recommended package manager)

### Installation

# Clone the repository  
git clone [repository-url]  
cd lyra-companion  
  
# Install dependencies  
pnpm install  
  
# Start development server  
pnpm run dev  
  
# Build for production  
pnpm run build

### Development Features

* **Hot Module Replacement**: Instant updates during development
* **TypeScript Support**: Full type checking and IntelliSense
* **ESLint Integration**: Code quality and consistency
* **Error Boundaries**: Graceful error handling
* **Performance Monitoring**: FPS and status indicators

## 🎨 Design Philosophy

### Cosmic Aesthetics

* **Color Palette**: Deep blues, purples, cyan accents
* **Visual Hierarchy**: Clear focus points and information flow
* **Sophisticated Simplicity**: Clean design with thoughtful details
* **Emotional Design**: Interface evokes wonder and connection

### User Experience

* **Intuitive Navigation**: Self-explanatory interface elements
* **Responsive Feedback**: Visual and audio confirmation of actions
* **Accessibility**: Keyboard navigation and screen reader support
* **Performance First**: Optimized for smooth 60fps experience

## 🚀 Performance Optimizations

* **Code Splitting**: Dynamic imports for optimal loading
* **Asset Optimization**: Compressed textures and models
* **Efficient Animations**: RequestAnimationFrame-based rendering
* **Memory Management**: Proper cleanup of 3D resources
* **Error Recovery**: Graceful degradation when features unavailable

## 🌐 Browser Compatibility

* **Chrome 90+**: Full feature support
* **Firefox 88+**: Full feature support
* **Safari 14+**: Limited TTS support
* **Edge 90+**: Full feature support

### Feature Requirements

* **WebGL 2.0**: For 3D graphics
* **Web Speech API**: For text-to-speech (optional)
* **Web Audio API**: For lip-sync analysis (optional)
* **ES2020**: Modern JavaScript features

## 🎯 Future Enhancements

### Planned Features

* **VRM Model Loading**: Support for custom VRM character models
* **Advanced Lip-Sync**: Phoneme-based mouth animation
* **Custom Dialogue**: User-definable character responses
* **Voice Training**: Personalized speech patterns
* **AR Support**: Augmented reality character projection
* **Multi-language**: International language support

### Technical Improvements

* **WebXR Integration**: VR/AR compatibility
* **Advanced AI**: LLM integration for intelligent responses
* **Cloud Sync**: Cross-device settings synchronization
* **Analytics**: Usage patterns and optimization insights

## 📄 License

This project is developed as a demonstration of modern web technologies and interactive 3D applications.

## 🤝 Contributing

Contributions are welcome! Areas for improvement: - Character animation systems - Voice synthesis enhancements - UI/UX improvements - Performance optimizations - Accessibility features

## 📞 Support

For questions, issues, or feature requests, please refer to the project documentation or create an issue in the repository.

**Lyra** - Where technology meets personality in a cosmic digital experience. ✨