Deploy to Render - Step by Step Guide

🚀 Render Deployment (Full Multiplayer)

Prerequisites:

1. GitHub account
2. Your code pushed to GitHub

Step 1: Push to GitHub

```
git add .
git commit -m "Ready for Render deployment"
git push origin main
```

Step 2: Deploy on Render

1. Go to render.com
2. Sign up with your GitHub account
3. Click **"New"** → **"Web Service"**
4. Connect your GitHub repository
5. Select this project repository

Step 3: Configure the Service

Fill in these settings:

- **Name**: `minecraft-multiplayer-game`
- **Environment**: `Node`
- **Build Command**: `npm install`
- **Start Command**: `npm start`
- **Instance Type**: `Free` (for testing)

Step 4: Environment Variables (Optional)

- **PORT**: `3000` (Render will override this automatically)

Step 5: Deploy

- Click **"Create Web Service"**
- Wait for deployment (takes 2-3 minutes)
- Your game will be live at: `https://minecraft-multiplayer-game.onrender.com`

🎮 What You Get:

- ☑ Full multiplayer support

- ☑ Socket.io real-time communication
- ☑ Persistent server
- ☑ Auto-scaling
- ☑ HTTPS by default
- ☑ Free tier available

🔧 Troubleshooting:

If build fails:

1. Make sure `package.json` has correct start script
2. Check that all dependencies are listed

If game doesn't load:

1. Check the logs in Render dashboard
2. Make sure the server is listening on the right port

🌐 Your Game URLs:

- **Game**: `https://your-app-name.onrender.com`
- **API/Socket**: Same URL (Render handles routing)

💡 Pro Tips:

1. Free tier may "sleep" after 15 minutes of inactivity
2. Paid tier ($7/month) keeps it always running
3. You can set up custom domains later
4. **OPTIMIZED**: Game now loads super fast with minimal world generation!
5. **PERFORMANCE**: Reduced chunk loading for instant multiplayer connections

⚡ Performance Optimizations:

- **Faster world generation**: Only 3 block layers instead of 64
- **Smaller chunks**: 1 chunk radius instead of 3
- **Frustum culling**: Only renders blocks visible to camera
- **Occlusion culling**: Skips blocks completely hidden behind others
- **Smart updates**: Only re-culls when camera moves significantly
- **Reduced view range**: 15 blocks optimized with visibility culling
- **Instant loading**: No more waiting for massive world generation!

🎮 Advanced Rendering Features:

- **Dynamic visibility**: Blocks automatically hide when you look away
- **Performance scaling**: More blocks visible = better culling performance
- **Memory efficiency**: Only renders what you can actually see
- **Smooth movement**: Camera-based culling updates smoothly

Ready to deploy? Just follow the steps above! 🚀