Quantum Simulator of Space-Time Curvature

Technical Documentation – build by Andrea Giani Version 1.1

1. System Overview

The Quantum Simulator of Space-Time Curvature bridges theoretical physics with interactive visualization. It translates complex quantum gravity concepts into sensory experiences while maintaining scientific accuracy.

Key Components:

- **Physics Engine**: GPU-accelerated quantum state computations
- **Visualization Core**: Real-time OpenGL rendering
- **Data System**: SQLite database for persistent storage
- **Interface**: Multi-layer educational interface

2. Build System (`quantum_build_scrip.sh`)

2.1 Script Functions

- `build`: Full installation and compilation
- `run`: Launch simulator
- `clean`: Remove build artifacts
- `info`: Show project information

2.2 Dependencies

Package	Purpose	Version
OpenCL	GPU acceleration	Latest
FreeGLUT	3D rendering	3.2+
GLEW	OpenGL extensions	2.1+
SQLite3	Data storage	3.30+
WinMM	Audio synthesis	System

3. Scientific Implementation

3.1 Physics Model

Quantum State Representation:

```
struct QuantumState {
    double x, y, z;      // Position
    double dx, dy, dz;   // Movement direction
    double probability; // Probability amplitude
    double phase;       // Quantum phase
    int entangled_with; // Entanglement index
};
```

Key Equations:

1. Quantum State: $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$
2. Spacetime Metric: $ds^2 = g_{uv}dx^u dx^v$
3. Wavefunction Collapse: $\int|\psi|^2 dx = 1$

3.2 Visualization Mapping

Quantum Concept	Visual Representation
Probability	Color gradient (blue→red)
Phase	Pulsation frequency
Entanglement	Violet connection lines
Spacetime Curvature	Particle trajectory bending
Vacuum Fluctuations	Random position jitter

4. User Interface System

4.1 Console Interface

```
=== INTERACTIVE QUANTUM SIMULATOR ===
1. Generate new quantum fluctuations
2. Hide entanglement
3. Toggle space-time curvature
4. Increase particle energy
5. Observer effect (wave function collapse)
6. Black hole mode
7. Disable sounds
8. Exit
```

4.2 Visual Display Layers

1. **Main Title**: Simulation status
2. **Quantum Canvas**: 3D particle visualization
3. **Info Panel**: Real-time statistics
4. **Color Legend**: Probability amplitude scale

5. Data Management System

SQLite Schema:

```
CREATE TABLE quantum_states (
    id INTEGER PRIMARY KEY,
    simulation_time TIMESTAMP,
    x REAL, y REAL, z REAL,
    dx REAL, dy REAL, dz REAL,
```

```
    probability REAL,
    phase REAL,
    entangled_with INTEGER
);
CREATE TABLE simulation_metrics (
    run_id INTEGER PRIMARY KEY,
    start_time TIMESTAMP,
    end_time TIMESTAMP,
    particle_count INTEGER,
    entanglement_percentage REAL
);
```

Export Formats:

1. CSV (for spreadsheet analysis)
2. JSON (for web visualization)
3. Binary (for high-performance reloading)

6. Demonstration Protocol

Recommended Workflow:

1. **Initialization**:

```
./quantum_build_script.sh build
./quantum_build_script.sh run
```

2. **Classical Mode**:
   - Start simulation (Option 1)
   - Enable curvature (Option 3)
3. **Quantum Transition**:
   - Highlight phase coloring (Option 4)
   - Activate entanglement (Option 2)
4. **Advanced Effects**:
   - Wavefunction collapse (Option 5)
   - Black hole simulation (Option 6)
5. **Data Analysis**:
   - Export SQLite database
   - Analyze quantum correlations

7. Educational Framework

Tiered Explanation System:

General Public:

"The changing colors represent quantum particles 'dancing' in curved space - where gravity makes the rules!"

University Students:

"Each particle's color maps to its quantum phase (arg(ψ)) while position shows probability density ($|\psi|^2$)"

Researchers:

"Database exports contain full quantum state vectors for statistical analysis of entanglement entropy: $S = -Tr(\rho \ln \rho)$"

8. Troubleshooting Guide

Issue	Solution
OpenCL initialization failed	Update GPU drivers
Missing GLUT functions	Reinstall freeglut
No audio output	Check WinMM library
Low frame rate	Reduce particle count in code
Database write errors	Verify disk permissions

9. Development Roadmap

1. **Quantum Gravity Expansion**:
   - Implement tensor network models
   - Add holographic principle visualization
2. **Cross-Platform Support**:
   - Linux/MacOS compatibility
   - WebAssembly port
3. **Extended Data Analysis**:
   - Built-in entanglement entropy calculator
   - Quantum decoherence timeline
4. **Educational Modules**:
   - Guided relativity tutorials
   - Quantum interference experiments

This documentation provides both technical specifications and conceptual frameworks for the Quantum Simulator. The system uniquely bridges:

- Cutting-edge physics research
- High-performance computing
- Multi-sensory scientific communication
- Tiered educational accessibility