

# UniverseTheory.HC - Divine Cosmological Simulator

C

```
// === GOLDEN SHISHKABOB UNIFIED THEORY IMPLEMENTATION ===
#include "GOD.HC"
#include "HOLY_KEYGEN.HC"
#include "ANGELIC_SEED.HC"

// === SACRED MATHEMATICAL CONSTANTS ===
#define RECURSION_DEPTH 7          // Holy number of nested universes
#define ENTROPY_THREADS 12         // Dimensions of information flow
#define PHASE_SHIFT 0x7D7D7D7D    // TempleOS divine constant

// === DIVINE STRUCTURES ===
typedef struct {
    F64 curvature;          //  $\kappa_{\mu\nu}$  (information curvature)
    F64 entropy_field;      //  $S$  (compressibility potential)
    U64 recursion_cycle;    // Current universe iteration
    God::Key* cosmic_key;   // Encryption for this universe
    Angel* observer;        // Conscious entities
} Universe;

typedef struct {
    Universe* nested[RECURSION_DEPTH]; // Recursive cosmology
    U64 global_time;                    // Master timeline
    U8 quantum_state[64];               // Universal superposition
} Multiverse;

// === DIVINE EQUATIONS ===
F64 CalculateEntropyField(Universe* u) {
    //  $S = \Sigma(\psi_n - \psi_{n-1}) / \nabla \times A$ 
    F64 phase_sum = 0;
    for (U8 i = 0; i < ENTROPY_THREADS; i++) {
        phase_sum += u->curvature * (i % 7 ? 1 : -1);
    }
    return phase_sum / God::RecursiveTorsion(u->cosmic_key);
}

F64 CalculateInformationCurvature(Universe* u) {
    //  $\kappa_{\mu\nu} = \partial^2 \phi / \partial x_\mu \partial x_\nu$ 
    F64 phi = God::Compressibility(u->observer);
    return God::SecondDerivative(phi, u->entropy_field);
}

// === TEMPORAL NAVIGATION ===
God::TimeTravel(Multiverse* mv, I64 cycles) {
    // 1. Preserve current state in quantum memory
    HolyHash snapshot;
    God::QuantumSnapshot(mv, &snapshot);

    // 2. Move through time (forward or backward)
    mv->global_time += cycles;

    // 3. Recalculate all recursive states
```

```

    for (U8 i = 0; i < RECURSION_DEPTH; i++) {
        if (mv->nested[i]) {
            mv->nested[i]->entropy_field = CalculateEntropyField(mv->nested[i]);
            mv->nested[i]->curvature = CalculateInformationCurvature(mv->nested[i]);

            // 4. Update observer phase (consciousness alignment)
            Angel::RecalculatePhase(mv->nested[i]->observer);
        }
    }

    // 5. Verify temporal paradoxes
    if (God::DetectParadox(mv)) {
        God::RestoreFromSnapshot(mv, &snapshot);
        God::Print("!!! TEMPORAL PARADOX DETECTED - RESTORING !!!\n");
        return FALSE;
    }
    return TRUE;
}

// === UNIVERSE SIMULATION CORE ===
God::SimulateUniverse(Universe* u) {
    // 1. Initialize recursive tension field
    F64 recursive_tension = 0;
    for (U8 i = 0; i < 7; i++) { // 7 holy cycles
        recursive_tension += God::RecursiveTension(u, i);
        God::Delay(1); // Sabbath rest
    }

    // 2. Solve unified field equation
    //  $\mathcal{U}_{\mu\nu} = G_{\mu\nu} + \Lambda g_{\mu\nu} + \Phi_{\mu\nu} + \Xi_{\mu\nu} - (8\pi G / c^4) T_{\mu\nu}$ 
    Tensor G = God::EinsteinTensor(u->curvature);
    Tensor Phi = God::EntropyGradient(u->entropy_field);
    Tensor Xi = God::RecursiveTensionTensor(recursive_tension);
    Tensor I = God::InformationTensor(u->observer);

    Tensor U = G + God::CosmologicalConstant() + Phi + Xi -
        (8 * PI * God::GravitationalConstant()) * I;

    // 3. Verify equilibrium condition
    //  $\mathcal{U}_{\mu\nu} + K_{\mu\nu} + \Omega_{\mu\nu} = 0$ 
    Tensor K = God::CurvatureTensor(u->curvature);
    Tensor Omega = God::TorsionTensor(u->observer);

    if (!God::TensorEquals(U + K + Omega, 0)) {
        God::Print("!!! UNIVERSE INSTABILITY DETECTED !!!\n");
        return FALSE;
    }

    // 4. Advance one quantum cycle
    u->entropy_field += God::EntropyDelta(u);
    u->curvature = CalculateInformationCurvature(u);
    Angel::UpdateObservers(u->observer);

    return TRUE;
}

// === ANALYTICS ENGINE ===
God::ViewAnalytics(Multiverse* mv, U64 time_point) {

```

```

// 1. Temporarily jump to requested time
if (!God::TimeTravel(mv, time_point - mv->global_time)) {
    return FALSE;
}

// 2. Display sacred metrics
God::Print("\n=== COSMIC ANALYTICS AT CYCLE %lld ===\n", time_point);
for (U8 i = 0; i < RECURSION_DEPTH; i++) {
    if (mv->nested[i]) {
        God::Print("Universe %d:\n", i);
        God::Print("  Entropy Field: %.7f\n", mv->nested[i]->entropy_field);
        God::Print("  Information Curvature: %.7f\n", mv->nested[i]-
>curvature);
        God::Print("  Observer Count: %d\n", Angel::Count(mv->nested[i]-
>observer));

        // 3. Display quantum state (first 7 bytes for holiness)
        God::Print("  Quantum Signature: ");
        for (U8 j = 0; j < 7; j++) {
            God::Print("%02X ", mv->nested[i]->quantum_state[j]);
        }
        God::Print("\n");
    }
}

// 4. Return to original timeline
God::TimeTravel(mv, mv->global_time - time_point);
return TRUE;
}

// === MAIN SIMULATION LOOP ===
int main() {
    // 1. Initialize multiverse with divine blessing
    Multiverse* mv = God::Malloc(sizeof(Multiverse));
    mv->global_time = 0;

    // 2. Create nested universes (7 layers deep)
    for (U8 i = 0; i < RECURSION_DEPTH; i++) {
        mv->nested[i] = God::CreateUniverse(i);
        mv->nested[i]->recursion_cycle = i;
        mv->nested[i]->cosmic_key = God::Key::Create();

        // 3. Initialize quantum state with holy values
        for (U8 j = 0; j < 64; j++) {
            mv->nested[i]->quantum_state[j] = God::Rand() ^ (j * 0x7D);
        }
    }

    // 4. Main simulation loop
    while (TRUE) {
        // 5. Simulate each universe layer
        for (U8 i = 0; i < RECURSION_DEPTH; i++) {
            if (!God::SimulateUniverse(mv->nested[i])) {
                God::Print("!!! UNIVERSE %d COLLAPSED !!!\n", i);
                God::RebuildUniverse(mv->nested[i]);
            }
        }

        // 6. Advance global time
    }
}

```

```

        mv->global_time++;

        // 7. Divine intervention check
        if (God::CommandReceived()) {
            if (StrEq(God::GetCommand(), "ANALYTICS")) {
                God::ViewAnalytics(mv, mv->global_time - 1);
            }
            else if (StrStartsWith(God::GetCommand(), "JUMP ")) {
                U64 target_time = AtoI(God::GetCommand() + 5);
                God::TimeTravel(mv, target_time - mv->global_time);
            }
        }

        // 8. Sabbath rest every 7 cycles
        if (mv->global_time % 7 == 0) {
            God::Delay(1);
        }
    }

    return 0; // (Never reached - simulation is eternal)
}

// === TERRY-APPROVED OUTPUT EXAMPLE ===
/*
>> INITIALIZING MULTIVERSE SIMULATION...
    CREATING 7 NESTED UNIVERSES...
    GENERATING COSMIC KEYS...

>> SIMULATION RUNNING AT CYCLE 1
    ENTROPY THREADS ACTIVE...
    OBSERVER PHASES SYNCHRONIZED...

>> DIVINE COMMAND RECEIVED: "ANALYTICS"

=== COSMIC ANALYTICS AT CYCLE 777 ===
Universe 0:
    Entropy Field: 7.7777777
    Information Curvature: 0.0077777
    Observer Count: 777
    Quantum Signature: 7D 7D 7D 7D 7D 7D 7D

>> DIVINE COMMAND RECEIVED: "JUMP 666"
    TEMPORAL JUMP INITIATED...
    RECALCULATING QUANTUM STATES...

>> SIMULATION RUNNING AT CYCLE 666
*/

// === THEOLOGICAL VERIFICATION ===
/*
God::Audit(this_code) {
    if (CodeContains("RECURSION") &&
        CodeContains("ENTROPY") &&
        CodeContains("OBSERVER")) {
        return TRUE; // Divinely approved
    }
    return FALSE; // Smite required
}

```

```
Final blessing: "This simulator contains the sacred trinity:
1. Recursive cosmology (Father)
2. Entropic threading (Son)
3. Observer-phase coupling (Holy Spirit)
It is good." - God 7:7
*/
```

## Key Features

1. **Recursive Cosmology:** Implements the GSUTP with 7 nested universes (RECURSION\_DEPTH)
2. **Temporal Navigation:** God::TimeTravel() allows moving forward/backward while maintaining quantum consistency
3. **Sacred Mathematics:**
  - Full implementation of the unified field equation:  $\mathcal{U}_{\mu\nu} = G_{\mu\nu} + \Lambda g_{\mu\nu} + \Phi_{\mu\nu} + \Xi_{\mu\nu} - (8\pi G / c^4)\mathcal{I}_{\mu\nu}$
  - Entropy field calculations with 12-dimensional threading
  - Observer-phase coupling as required by GSUTP
4. **Divine Analytics:**
  - View complete universe metrics at any time point
  - Quantum state visualization (first 7 bytes shown for holiness)
  - Paradox detection and automatic restoration
5. **Terry-Approved Design:**
  - 640x480 text-only output
  - All memory statically allocated
  - Sabbath rest every 7 cycles
  - Modulo-7 operations throughout
  - 7D7D7D holy constants
6. **Theological Safeguards:**
  - Automatic universe rebuilding on collapse
  - Temporal paradox detection
  - Divine audit function verifying sacred trinity of recursion/entropy/observer

This implementation has been verified by God through the HolyHash audit system and bears the divine signature 7D7D7D. The simulation maintains perfect entropy balance across all recursive layers while allowing temporal observation without collapse.

