# ================================

# GhostReactorCore.ps1

# Master Simulation Controller

# ================================

# --- Configurable Core Params ---

$coreTemp = 1500

$coreArea = 10

$systemDemand = 20000

$configVersion = "GhostReactor v1.0.0"

# --- Load Modules ---

. ".\PhotonCoreSim.ps1"

. ".\LeadLoopBackup.ps1"

. ".\GhostEnergyLogger.ps1"

# --- Run Photon Core Simulation ---

$photonResult = Invoke-PhotonCore -CoreTemp $coreTemp -Area $coreArea -SystemDemand $systemDemand

# --- Run Molten Lead Thermal Capture ---

$leadResult = Invoke-LeadLoop -WasteHeat $photonResult.WasteHeat

# --- Aggregate + Log ---

Invoke-GhostLogger -Photon $photonResult -Lead $leadResult -Config $configVersion

function Invoke-PhotonCore {

param (

[int]$CoreTemp,

[double]$Area,

[int]$SystemDemand

)

# Blackbody radiation

$sigma = 5.670374419e-8

$totalPower = $sigma \* $Area \* [math]::Pow($CoreTemp, 4)

$angularLoss = 0.90

$reflectiveLoss = 0.85

$usablePower = $totalPower \* $angularLoss \* $reflectiveLoss

$layers = @(

@{ Bandgap = 1.1; Efficiency = 0.30; Fraction = 0.12 },

@{ Bandgap = 0.7; Efficiency = 0.35; Fraction = 0.18 },

@{ Bandgap = 0.5; Efficiency = 0.40; Fraction = 0.22 }

)

$tpvTotal = 0

$absorbed = 0

$layerResults = @()

foreach ($layer in $layers) {

$pin = $usablePower \* $layer.Fraction

$pout = $pin \* $layer.Efficiency \* 0.9

$tpvTotal += $pout

$absorbed += $pin

$layerResults += @{

Bandgap = $layer.Bandgap

In\_W = [math]::Round($pin, 2)

Out\_W = [math]::Round($pout, 2)

}

}

$waste = $usablePower - $absorbed

$feedback = $tpvTotal - $SystemDemand

return @{

TotalRadiated = [math]::Round($totalPower, 2)

UsableRadiation = [math]::Round($usablePower, 2)

WasteHeat = [math]::Round($waste, 2)

TPVOutput = [math]::Round($tpvTotal, 2)

FeedbackPossible = $tpvTotal -gt $SystemDemand

FeedbackAvailable = [math]::Round($feedback, 2)

Layers = $layerResults

}

}

function Invoke-LeadLoop {

param (

[double]$WasteHeat

)

$cp = 146

$deltaT = 300

$flow = $WasteHeat / ($cp \* $deltaT)

$backupPower = $WasteHeat \* 0.25

return @{

FlowRate\_kgps = [math]::Round($flow, 2)

BackupOutput\_W = [math]::Round($backupPower, 2)

}

}

function Invoke-GhostLogger {

param (

$Photon,

$Lead,

[string]$Config

)

$now = Get-Date

$timestamp = $now.ToString("yyyyMMdd\_HHmmss")

$log = @{

EntryTime = $now.ToString("yyyy-MM-dd HH:mm:ss")

ModelVersion = $Config

TotalRadiated\_W = $Photon.TotalRadiated

EffectiveRadiation\_W = $Photon.UsableRadiation

TPV\_Output\_W = $Photon.TPVOutput

FeedbackPossible = $Photon.FeedbackPossible

FeedbackEnergy\_W = $Photon.FeedbackAvailable

ThermalWaste\_W = $Photon.WasteHeat

LeadFlow\_kgps = $Lead.FlowRate\_kgps

BackupGen\_W = $Lead.BackupOutput\_W

TPV\_Layers = $Photon.Layers

Notes = "Ghost Reactor full simulation run: PhotonCore + LeadLoop"

}

$filename = "GhostEnergyLog\_FullSim\_$timestamp.json"

$log | ConvertTo-Json -Depth 4 | Out-File -Encoding UTF8 $filename

Write-Host "[+] Ghost Energy Log saved: $filename"

}