SmartCitySim – AI-Powered Sustainable City Simulation Game (Design Document)

Title Page

Game Title: SmartCitySim – AI-Powered Sustainable City Simulation Game
Version Number: 1.0
Date: September 15, 2025
Team / Developer:

Lead Developer: Valerie Tan Ying Ying

Department of Computer Science & Game Development

Dongseo University

Table of Contents

D. Usability Study Protocol

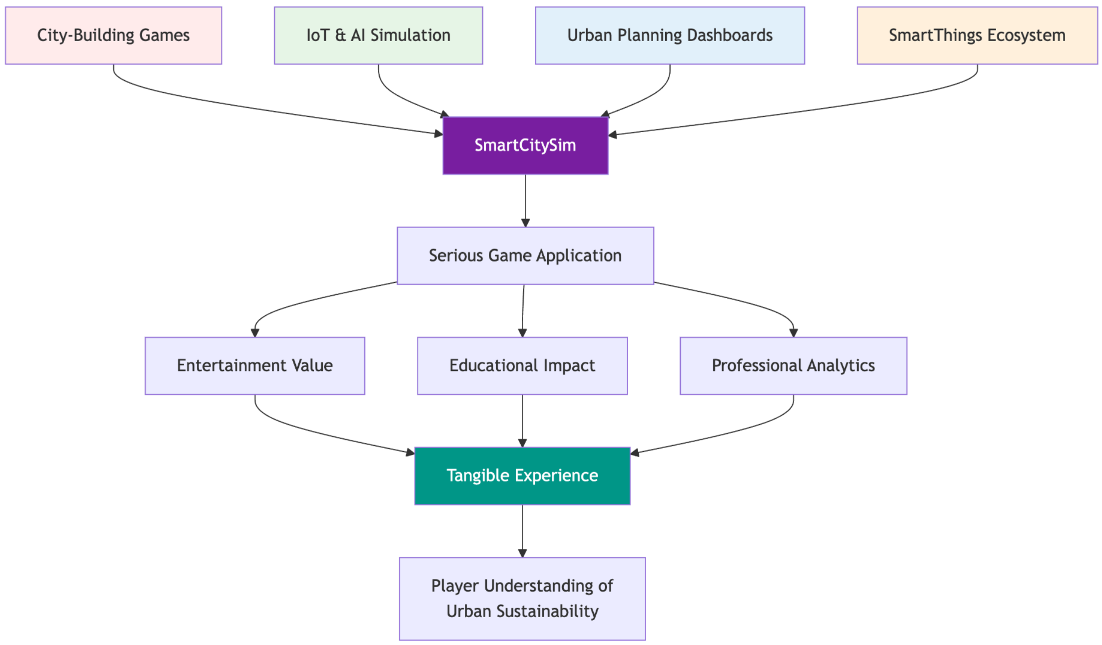
Executive Summary / Project Overview

SmartCitySim is a **serious game** that blends **Sims/SimCity-inspired city-building mechanics** with **real-time sustainability analytics**. Players act as **urban planners**, making choices that impact **traffic flow, CO_2 emissions, energy use, flood risk, and citizen happiness**.

The game integrates **data-driven dashboards** with an **interactive city builder**:

- **Game Layer:** Players place roads, buses, solar panels, and parks.
- **Simulation Layer:** Predictive models estimate sustainability metrics.
- **IoT Tie-In:** A SmartThings-inspired system simulates how **household-level actions** (e.g., reducing home energy use) scale to **city-wide impact**.

By uniting **AI/ML, IoT concepts, and gamification**, SmartCitySim both **educates players** and **demonstrates smart city platform potential**, aligning with initiatives from **Samsung SDS & Samsung C&T**.



See Figure 1: System Architecture Diagram

Motivation & Problem Statement

Context: Urban areas face escalating **traffic congestion, climate change, and energy inefficiency**. While **smart city platforms** exist, citizens often lack **engaging tools** to understand the **consequences of their choices**.

Problem: Current smart city dashboards are **technical** and not designed for citizen engagement. Games like *SimCity* are **engaging**, but **lack real-world sustainability integration**.

Solution: SmartCitySim bridges this gap by merging **serious data dashboards** with **playful, Sims-like city-building**. It makes **sustainability trade-offs visible** and **interactive**, motivating both **learning** and **behavioral awareness**.

Impact:

- **Education:** Helps users understand trade-offs between transport, climate, and livability.

- **Research:** Serves as a prototype for integrating **IoT + AI + gamification** in urban planning.

Industry Alignment: Matches smart city R&D focus areas of Samsung SDS and related organizations.

Target Audience and Personas

Primary Users:

Students & Citizens: Learn sustainability through gamified simulation.

Gamers: Fans of *SimCity* and *management sims*.

Policy Students: Use tool for urban planning scenarios.

Secondary Users:

Smart City Researchers: Prototype for Samsung SDS, government agencies.

Educators: Classroom tool for sustainability awareness.

Personas:

Student Planner (Age 21): Wants hands-on sustainability training.

Eco-Gamer (Age 25–35): Enjoys simulations, experiments with CO_2 reduction strategies.

Analyst (Age 30+): Uses tool to demo IoT + energy-saving effects on cities.

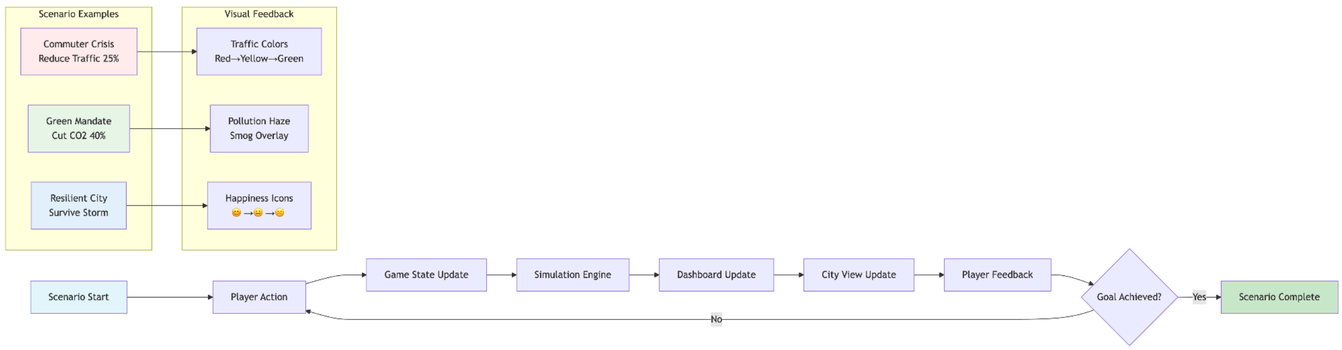
Gameplay / User Journey Scenarios

Traffic Optimization: Player adds buses → congestion & CO_2 drop.

Flood Resilience: Player builds parks → flood risk decreases, happiness rises.

IoT Household Action: Player reduces household energy via SmartThings → CO_2 reduction city-wide.

Trade-Offs: Building solar farms cuts emissions but costs resources → players must balance.




```
Figure 2: Scenario Gameplay Flow Diagram
```

Core Features Overview

7.1 City Builder Mechanics

Tile-based city grid (roads, buildings, green spaces).

Place, upgrade, demolish infrastructure.

Citizens & vehicles simulated as agents.

7.2 Sustainability Metrics Dashboard

Metrics: **Traffic, CO_2, Energy, Flood Risk, Happiness**.

Heatmaps + graphs for real-time visualization.

7.3 Traffic & Energy Simulation

Rules-based for baseline.

ML models for predictions (traffic congestion, emission curves).

7.4 SmartThings IoT Tie-In

Simulated IoT device scaling.

Example: “Turn off 10,000 AC units → -2% CO_2 emissions.”

7.5 Gamification

Happiness Meter.

Unlockable upgrades: EV stations, bike lanes.

Leaderboard: Compare sustainability performance.

User Interface (UI) & Accessibility

Two Views:

- **Gamified City View** (tilemap city builder).
- **Professional Dashboard View** (charts, heatmaps).

UI Principles:

- High-contrast visuals.
- Clear color-coded sustainability indicators.
- Simple interaction flow: build → see impact.

System Architecture

9.1 Unity Frontend (Tilemap & Agents)

Tile-based 2D/3D city grid.

Agent-based cars/citizens.

9.2 Backend Simulation & ML API

Framework: Python Flask REST API.

Function: Receives city layout JSON, returns simulation results.

Scalability: Designed for potential migration to FastAPI for enhanced performance.

9.3 Database (PostgreSQL/PostGIS + Redis)

Stores layouts, simulations, metrics.

Redis cache for performance.

9.4 IoT Simulation Layer

Mock SmartThings API.

Household → district → city scaling model.

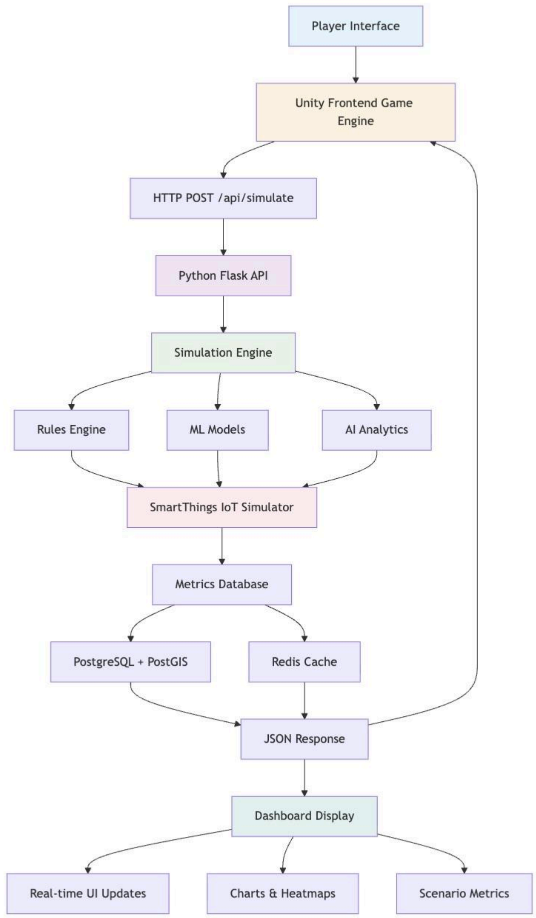


Figure 3: System Architecture Overview

Machine Learning Architecture

10.1 Rules-Based Baseline

Example: +100 buses = -5% congestion.

10.2 Predictive ML Models

Regression & LSTMs for congestion & emissions.

Synthetic + open datasets (Seoul Open Data, OSM).

10.3 Fusion Pipeline

Ensemble: traffic + energy + climate.

Outputs combined sustainability score.

10.4 Model Selection Strategy

Scikit-learn: Linear regression, decision trees for baseline models

TensorFlow/Keras: LSTM networks for time-series traffic prediction

PyTorch: Custom neural networks for complex urban simulations

Ensemble approach:Combine simple and complex models for robustness

---

App Modules & Data Flow

Flow:
Unity Client → Backend API → ML Engine → Metrics DB → Unity/Dashboard UI.

Modules:

Tilemap Editor (city building).

Agent Simulation (traffic flow).

ML Prediction Engine.

Dashboard Visualization.

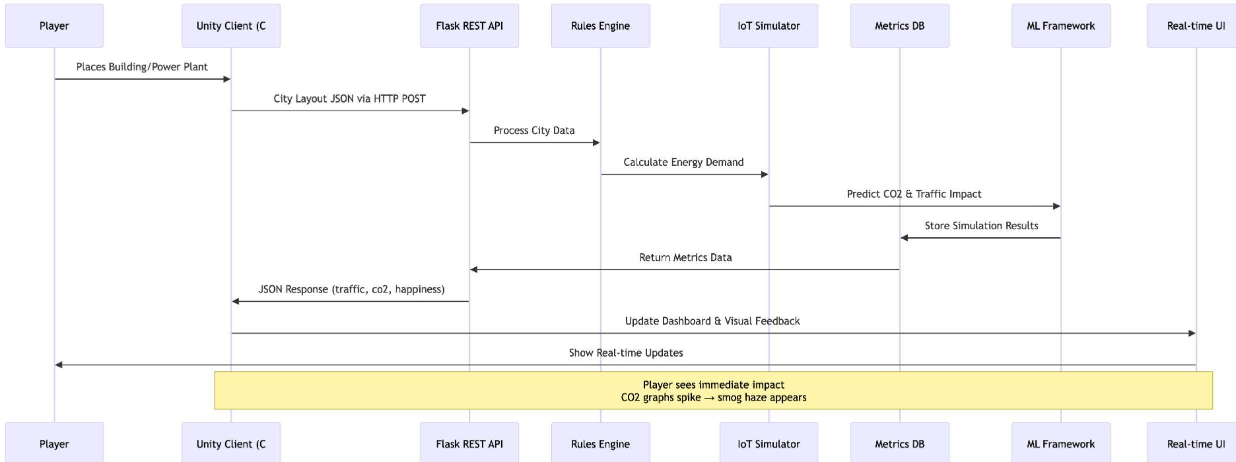


Figure 4: Real-time Implementation Sequence

Development Roadmap (12-Month)

MVP Path (Weeks 1–12):

Unity city-builder skeleton.

Rules-based traffic sim.

Dashboard with live metrics.

IoT mock actions.

Advanced ML Path (Weeks 13–24):

ML congestion predictor.

Climate & energy models.

IoT scaling.

Final Phase (Weeks 25–26):

Usability testing.

UI/UX polish.

Final report & demo.

Testing & Validation Plan

13.1 Usability Testing:

Recruit 5–10 players.

Evaluate clarity of dashboard/game link.

13.2 Simulation Benchmarking:

Compare congestion predictions with open datasets.

13.3 Offline Robustness:

Test Unity-client demo mode without backend.

Technology Stack & Platforms

Frontend (Game): Unity (C#) with Tilemap system for city building.

Backend: Python Flask with REST API + Machine Learning models.

Database: PostgreSQL + PostGIS (spatial data), Redis (caching).

Simulation: Hybrid rules-based + ML predictive models.

ML: TensorFlow/Keras or PyTorch.

Deployment: Docker containerization, AWS/Azure cloud deployment.

IoT Integration: SmartThings-inspired simulated API.

Budget & Resource Planning

Resource	Cost (USD)	Justification
Unity Assets	$150	Tilemaps, models.
Datasets	Free	Open data portals.
Cloud Hosting	$100	Backend deployment.
Dev Licenses	$25	GitHub Student/Play Store.
Contingency	$200	Unexpected costs.

Total Estimate: ~$475

Risk Management

Data Gaps: Use synthetic data if real unavailable.

Performance Issues: Optimize with Redis caching.

Over-Complex Scope: MVP first, ML later.

Integration Risks: SmartThings tie-in simulated, not live.

Appendices

A. Diagrams:

System architecture diagram.

ML pipeline flow.

Roadmap Gantt chart.

B. Sample Data:

Seoul traffic dataset.

Synthetic CO_2 reduction samples.

C. External References:

OSM, Seoul Open Data, IPCC emission factors.

D. Usability Study Protocol:

Playtest feedback forms.

Performance benchmarks.