Project on

Metro Mayhem

for

Software Project Management

Bachelor of Technology in Computer Science Engineering

B. M. Institute of Engineering & Technology, Sonipat

Submitted to:
Ms. Sonika Vasesi

Submitted by:
Bharat Mishra (CSE/22/139)

Abstract

Metro Mayhem is an open-world gaming experience that aims to recreate the vibrant atmosphere of India's capital city. Players can explore accurately modeled neighborhoods, drive various vehicles including cars and auto-rickshaws, and interact with a dynamic environment that reflects the city's unique culture and daily life. The game features realistic traffic patterns, day-night cycles, and AI-driven crowd systems that mirror the authentic hustle and bustle of New Delhi.

Software Requirements

Development Tools

- Godot Engine 4.2 or later (Primary Game Engine)
- Blender 4.0 or later (3D Modeling and Animation)
- Visual Studio Code (Script Editor)
- Git (Version Control)
- GIMP/Krita (Texture Creation)
- Audacity (Sound Editing)

Runtime Dependencies

- Operating System: Windows 10/11, Ubuntu 20.04 or later, macOS 12 or later
- Graphics API: Vulkan or OpenGL 3.3+
- Additional Libraries:
    - OpenAL (Audio)
    - SDL2 (Input Management)
    - PhysX (Physics Engine built into Godot)

Hardware Requirements

Development System

- Processor: Intel i7/AMD Ryzen 7 or better
- RAM: 16GB minimum, 32GB recommended
- Graphics: NVIDIA GTX 1660 or better with 6GB VRAM
- Storage: 50GB SSD minimum
- Display: 1920x1080 resolution or higher

Target Platform (Player Requirements)

- Processor: Intel i5/AMD Ryzen 5 or equivalent
- RAM: 8GB minimum
- Graphics: NVIDIA GTX 1050 or equivalent with 4GB VRAM
- Storage: 25GB available space
- Internet: Broadband connection for potential future updates
- Input: Keyboard/Mouse, Gamepad support

Software Lifecycle Model

The project will follow an Agile-Scrum development methodology with weekly sprints:

1. Planning Phase (1 week)

Project scope definition, Technical requirement analysis, Asset list creation, Timeline establishment

2. Prototype Phase (2 weeks)

Core mechanics implementation, Basic environment setup, Vehicle physics testing, Initial performance optimization

3. Development Phase (5 weeks)

Iterative feature implementation, Regular playtesting, Continuous integration, Bug fixing and optimization

4. Polish Phase (2 weeks)

Visual improvements, Performance optimization, Bug fixing, Final testing

Modules

Core Game Systems

World Management System, Time and Weather System, Save/Load System, Performance Optimization System

Player Systems

Character Controller, Vehicle Controller, Camera System, Input Management, Inventory System

Vehicle Systems

Vehicle Physics, Traffic AI, Collision Detection, Damage System, Vehicle Customization

Environment Systems

Dynamic Weather, Day/Night Cycle, Traffic System, Crowd Simulation, Building Management

AI Systems

Pedestrian AI, Traffic AI, NPC Behavior System, Path Finding, Dynamic Events System

UI/UX Systems

Main Menu, HUD System, Map/Navigation, Settings Menu, Performance Metrics

Audio Systems

Environmental Audio, Vehicle Sounds, Ambient City Sounds, Music System, Dynamic Audio Mixing

Graphics Systems

Lighting System, Shader Management, Particle Effects, LOD System, Post-processing Effects