City Builder System - Report Brief

https://docs.google.com/document/d/103Nhj6JCw0WtIQdYZTP1Hnx0jN7RK9INIYK_mzBfss0/edit?usp=sharing

Core Urban Development Components

• Transportation needs and traffic flow
• Public utilities (power, water) management
• Housing requirements for citizens
• Healthcare and education facilities
• Government policies and tax
• Budget allocation and financial constraints
• Economic growth indicators
• Population welfare metrics
• Crime and punishment

Design Patterns and Implementation

Infrastructure Management: The Factory Method pattern is used in the `BuildingFactory` class to simplify the creation of various types of buildings, such as houses, factories, offices and more. The State design pattern is used to manage the utilities of each building.

Economic Systems: The Observer pattern is utilized in the `CitizenObserver` and `CityObserver` classes. This allows the city manager to monitor and respond to changes in citizen welfare metrics, such as healthcare, education, and crime rates, ensuring the necessary reponses are made. The Memento design pattern is also used to store and compare city metrics to previous years, ensuring economic growth over time.

Social Services: The State pattern and Chain of Responsibilty patterns are used in the `CitizenMood` and `ComplaintHandler` classes to monitor citizen moods and complaints. This ensures that social welfare is met, by monitoring the needs of citizens.

Crime Punishment: The Strategy pattern is implemented in the `CrimePunishmentStrategy` class, enabling the dynamic adjustment of crime punishment, so that the punishment fits the crime. Ranging from community service to a death sentence.

Metric Monitoring: The Visitor design pattern is used to generate reports on the city metrics and structures, calculating various statistics based on the `City` class.

Menu Management: The Command design pattern is used to manage the functionality behind the user interface and allow the client to make a range of requests regarding city management, such as changing policies, adding buildings, exiting the program, etc. The Command design pattern manages the functionality of the other design patterns.

Taxation: The Command design pattern is used in the `AdjustTaxCommand` class to modify the tax state of the city. The Singleton pattern is used in the `TaxSystem` class, ensuring a centralized point of control and coordination for the tax needs of the city.

The Template Method pattern is utilized in the `Visitor` class and its subclasses, defining a common structure and workflow for generating various types of city reports, while allowing for customization in the specific reporting logic.

By leveraging these 10 design patterns, our city builder program maintains a high degree of modularity, flexibility, and scalability. This architectural approach enables us to effectively manage the intricate relationships and dynamics of the urban environment, while also providing a foundation for future expansion and adaptation to evolving city management requirements.

References

Urban social functional requirements for a 20-minute neighbourhood in Melbourne: https://www.researchgate.net/figure/Urban-social-functional-requirements-for-a-20-minute-neighbourhood-in-Melbourne_fig2_361407207

How to build a city, step-by-step: A DIY guide: https://www.theguardian.com/cities/2015/jun/30/how-build-city-step-by-step-diy-guide

Taxes: https://www.investopedia.com/terms/t/taxes.asp