# Specification General Rundown Of how our City Builder Simulation works:

Progress/Downfall of a city will be simulated over a couple of year

City will contain:

Buildings (Residential, Commercial, Industrial, Landmarks)

Utilities (Power plants, water supply, waste management, sewage systems)

Transportation (Roads, Public transit, Trains, Airports)

Citizens (Population growth, Employment, Services, satisfaction)

Government (Taxation, City Budget, Policies, Public Services)

Resources (Materials, Energy, Water, Budget)

Taxes (Tax rates, Collection, Allocation, Impact)

City Growth (Population Growth, Housing Needs, Economic development, Infrastructure development)

Start of simulation -> predefined parameters for the city

BUILDINGS:

* Design Pattern: Abstract factory to create the buildings which each belong in a family
* Starting point: a few of each building I guess
* User ability:
  + Buy buildings: will diminish city budget and use resources
  + Destroy buildings: will add to materials resource
* Effect of buildings:
  + **Residential**: decide the maximum population capacity
  + **Commercial**: will affect future city budget
    - More commercial buildings -> generate income over years for city budget obtained from tax
  + **Industrial:** decide maximum resource capacity ( even though industrial buildings take up resources, that very same resource will be boosted due to production ( so building a warehouse will take concrete but will boost wood capacity by 0.25%)
    - **Factories:** Affect total steel
    - **Warehouses:** Affect total wood which is used to create tar as well as total concrete
    - **Plants:** Affect total power
    - **Dams:** Affects total water ( newly added )
  + **Landmarks:** will affect citizen satisfaction
    - More the better
* General: Each building will have a certain cost (which will be subtracted from city budget) and resources required (which will subtract from resources) when built.
* Pattern Implementation:
  + Abstract Factory will be used to create buildings, every building built will result in a certain number of resources being used. This will be directly linked to resources via a pointer to a resource object.
* Classes:
  + BuildingFactory
  + ResidentialFactory
  + CommercialFactory
  + LandmarkFactory
  + Residential ( House, Flat , Townhouse , Estate) -> has a capacity and a resourceManager object which is instantiated in the constructor
  + Commercial ( Shops , Mall , Office ) -> has a government object in the class which affects budget. Also affects materials obviously.
  + Industrial ( Factories , Warehouse , Plants , Dams) -> Deals with ResourceManger by adding percentage increases in capacities and also taking resources for building costs.
  + LandMarks ( Parks , Monuments , Cultural centres) -> Statistics and ResourceManager objects which deal with satisfaction of clients as well as resource costs.
* Note that in each ConcreteProduct, the constructor will have specific values set for resources taken and will call the Resources object through the constructor to deduct resources and create building.

RESOURCES:

* Design patterns: Observer
  + When the resources drop below a certain threshold, observer makes calls to the utilities to enforce alternative strategies
  + E.g., when power drops below 10% -> activate nuclear & decrease satisfaction
    - When nuclear drops below 10% -> boom
  + E.g., When hygiene drops below 40% -> increase random chance of disease outbreak -> decrease satisfaction
  + Etc.
* Starting Point: predefined amount of starting resources including:
  + 10 000 steel
    - Needed to build power plant and other buildings
  + 15 000 concrete
    - Needed to build hospitals and other facilities
  + 5000 ~~tars~~ wood
    - Used to construct and upgrade roads as well as the construction of buildings
    - Instead of tar we will use wood since the wood can be used as building materials and as a resource used to create tar.
  + 10 000 water
    - Water will be measured in Kilo litres ( x10^3)
    - Water dropping below a certain amount will invoke a command which sends the city into a critical state, therefore endangering the end due to draught.
  + 50 000 power
    - Measured in KW
    - Dropping below a certain point will invoke observer which sends city into using nuclear power.
* User Ability:
  + User’s goal is to efficiently manage their resources to effectively build necessary requirements in a city
* Effect of parameters:
  + **Materials:** 
    - Same as mentioned above, there will be different resources that the user makes use of
    - Depending on the city’s satisfaction at the end of each year, they are rewarded with an amount of resources directly proportional to their citizen’s satisfaction
  + **Energy:** 
    - Will be produced by the power plant
    - Necessary for running of buildings and a lack of causes dissatisfaction along with potential boom
  + **Water:**
    - Will be produced by water supply
    - Necessary for citizens to live, lack of water will increase death rate
    - Can also cause drought if water runs out
  + **Budget:**
    - Same thing as city budget
* Pattern Implementation:
  + The subject will be ResourceManager, and its concrete subjects will be:
    - ConcreteWoodManager
    - SteelManager
    - PowerManager
    - WaterManager
  + If one of these managers enter a critical state, a notify() will be sent to the command pattern in Utilities.
  + This means that CommandInvoker will also be the observer class for the observer pattern.
  + Therefore CommandInvoker is an invoker for the command pattern as well as the Observer for the observer design pattern.
  + Observers changes, if notified, updates accordingly by selecting the correct command which is then sent back to the concreteSubjects.

UTILITIES:

* Design Pattern: Command and Adapter ( Leave adapter out for now, implementation might complicate things unnecessarily)
* Starting point: one of each plant I guess
* User Ability:
  + Buy utilities: will diminish city budget and use resources
  + Destroy utilities: will add to materials resource
* Effect of buildings:
  + **Power plants:** generate electricity
    - More buildings = higher energy consumption
    - Create jobs for `energyWorkers`
  + **Water supply:** generate water
    - More residential houses and landmarks = higher water consumption
    - Create jobs for `waterWorkers`
  + **Waste management:** decide level of hygiene
    - More citizens = more waste management required
    - More industrial buildings = more waste management required
    - Create jobs for `wasteWorkers`
  + **Sewage systems:** decide level of hygiene
    - More citizens = more sewage systems required
    - Create jobs for `sewageWorkers`
* General:
  + Essential for functioning of the city -> if not looked after will be end of city
  + Power:
    - More buildings -> more power consumption
    - When power drops too low -> citizen satisfaction diminished
    - Option to change over to nuclear -> if that drops too low then city explodes
  + Water:
    - More Resident & Landmarks buildings -> more water consumption
    - Water drops too low -> citizen satisfaction diminished -> maybe they drought and they all die
    - Have option to open damn wall to restock water (only get one)
  + Waste & sewage:
    - If drops too low -> change of outbreak and then chance everyone die
* Pattern implementation:
  + This pattern will have a bunch of command leading to either resources or transportation. The commands are a mixture of state changes and warnings meant to guide the user while building their city.
  + Some of these commands include:
    - WarnWaterLow
    - WarnPowerLow
    - ChangePowerOption
    - WarnWaterLow
    - DecreaseWaterConsumption
    - IncreaseHygiene
    - DecreaseHygiene
    - WarnMaterialsLow
  + Note that the invoker in this class is also known as the observer in the observer pattern. Invoker will have a bunch of Booleans which will trigger the correct command. ( e.g. if (LowEnergy==true) { initiate Nuclear Command } ) The selection of the correct command will be done using the update() method from the observer pattern.
  + The Receivers in this case will be the concreteSubjects in the observer pattern.

Complete remark on UTILITIES:

* Two of the things utilities is supposed to do is already being done in the resources class ( generation of electrity and water distribution) and so is there really any need to have a specific design pattern for utilities?
* Because same can be done for waste and sanitation aspects, no need for a specialised design pattern over here?

TRANSPORTATION:

* Design Patterns: strategy, state, observer
  + State -> state of roads
    - Not Operational:
      * Work comes to a halt (no power generation, etc. )
      * Decrease citizen satisfaction severely
    - Bad:
      * 25% reduced productivity (power generation)
      * Decrease citizen satisfaction slightly
    - Okay:
      * No pros or cons
    - Good:
      * 25% increased productivity
      * Increase citizen satisfaction slightly
    - Exceptional
      * 40% increased productivity
      * Increase citizen satisfaction severely
  + Observer -> when roads change states, make calls to productivity and citizen satisfaction
  + Strategy -> IDK
* Starting Point: IDK, maybe start with basic roads and public transit
* User ability:
  + Users can opt to invest in trains, airports, upgrade roads, and upgrade public transit
  + Will cost resources and budget
* Effect of buildings:
  + **Roads**:
    - can contribute to citizen satisfaction (amenities)
    - Increase employer efficiency
  + **Public Transit:**
    - Same as above
  + **Trains:**
    - Same as above
  + **Airports:**
    - Can serve as a means of passive income from foreigners which contribute to the city budget. Will be expensive to build
* General:
  + Basically just options for the user to upgrade to help improve efficiency and satisfaction of citizens

CITIZENS:

* Design Patterns: Prototype, template
  + Template: Different types of citizens based on their job
    - energyWorker: will work for power supply
    - waterWorker: ditto
    - wasteWorker: ditto
    - sewageWorker: ditto
    - unemployed: when no jobs are available and population continues to grow
      * essentially provide nothing to city but increases citizen consumption and needs
    - homeless: basically like unemployed but occurs when no more resident buildings are available
  + Prototype: used to create instances of citizens
* Starting point: will have a pop of like 50 or something
* User Ability: No direct control of the citizens, they just have to model the city in such a way that they are satisfied and happy -> will basically be a measure of success
  + Do have control over the purchases of services (healthcare, security, entertainment)
  + Like before these will cost from budget and resources
* Effect of parameters:
  + **Population Growth:** 
    - Will use a population growth formula which is applied every year
    - Factors:
      * Birth Rate
        + Healthcare
        + Bad education
      * Death Rate
        + Lack of health care
        + Bad hygiene
        + Bad security
        + Lack of water
        + Bad satisfaction
  + **Employment:**
    - Will be decided by utilities
    - There will be available jobs and when this is exceeded, they become unemployed
  + **Services:**
    - Healthcare -> better healthcare = decrease death rate and increase birth rate. Can also be used in the deciding factor if the city succumbs to an outbreak
    - Education -> decrease birthrate – prevent population boom
    - Security -> decrease deathrate and increase satisfaction
    - Entertainment -> increase satisfaction
  + **Satisfaction:**
    - Calculated from all the factors which have been mentioned (I will make a specification which specifies everything which affects citizens’ satisfaction)
    - This is basically the score of the game, the user should aim to score for highest possible satisfaction

GOVERNMENT: (Ronan had some interesting ideas he could add here)

* Design Patterns: Mediator, Singleton
* Starting Point: A city budget of $1 000 000 or alike can be the starting point
* User Ability:
  + Control of budget
    - Use of budget (mainly for building)
    - Increase of budget (decided by tax rates, foreign income, etc.)
* Effect of parameters:
  + **Taxation:** will be elaborated on in taxes class
  + **City Budget**: Will be the pool of money in which the user uses to upgrade the city
    - It will be their responsibility to manage their budget efficiently and will be the main goal of the game, along with citizen satisfaction
  + **Policies:** Not too sure what we can add here which affects city dynamics
  + **Public Services:** Same as services for citizens, maybe we have some worker as follows:
    - healthcareWorker
    - lawEnforcement
    - teacher
    - etc
    - These will function similar to the other workers and will have to be implemented in the template design pattern for citizen
    - Will also need power and roads to remain functional

TAXES

* Design patterns: Strategy & memento
  + Strategy
    - Different ways to calculate taxation strategies
      * Progressive
      * Flat
      * Regressive
  + Memento
    - Previous years tax will be used to calculate current tax
* Starting point: User chooses their beginning tax rates and the strategy
* User ability:
  + Will have the option to change their tax rates each year
  + A high tax rate will yield in an increase in budget, however, will decrease citizen satisfaction
* Effect of parameters:
  + **Tax Rates**
    - Adjustable rates
    - Different types of tax
  + **Collection** 
    - Will be collected from workers – more workers more income tax for city budget
  + **Allocation** 
    - Not much to say here
  + **Impact** 
    - More tax – increase in city budget but decrease in satisfaction which leads to a decrease of material rewards

CITY GROWTH

* Design patterns: Memento
  + Memento used to store each years stats
  + Can then be used to calculate improvements or deterioration of current year compared to previous years
  + Basically users can check their stats from previous years
* Starting point:
  + Display the starting points stats
* User ability:
  + Not much user ability, just a summary of the year’s progression
* Effect of parameters:
  + **Population growth:** obtained from citizens population growth
  + **Housing needs:** comparison of citizens and residential buildings
  + **Economic Development:** Commercial and Industrial stats
  + **Infrastructure expansion:** road and utilities stats

ADDITIONAL FEATURES we can add if we have time:

<some ideas to be added here, I forgot but them now but will add when I remember>