

Functional Need:

The Main functional need of any project is the logic of the project. We can determine each person's particular role in the project and can easily determine the project's functional requirements. We also need to define the functional requirements during the gathering, analysis and planning phase, but before the modeling phase of the project lifecycle.

The functional need of this project would be creating region and configuration files and then initializing the simulation. The simulation we are trying to construct is a sim city simulation so it is important to have certain things to help with implementation. The process that we are trying to implement would be the simulation of simcity's growth over time as well as the growth of the residential, commercial, and industrial zones. The different functions of this functionality would be determining how each zone grows over time and how each zone connects to each other. The data and functions are going to require how many people have come to the city, how many houses and other residences are built, how many commercial buildings are there and potentially how much the job market has grown as well as how many industry and industry buildings have moved here. The functionality needs to occur with each part to create an image of how all these pieces connect to each other and how much the city has grown as a whole. Other components that require data from each other would be that each portion of the project requires data about itself but also from the city as a whole to be able to accurately show its growth and the growth of each part.

Data Storage:

For efficient data storage and management system in the project we need to check and maintain the following two things which are as follows:

Database architecture

Scope, Maintainability, and Reliability

Data architecture represents the type of architecture we need to design the data and the way in which we store data for this project. We need to have fast retrieval and be effective with our security. For project management we can implement a centralized system.

Scope, Maintainability and Reliability of the project depends on the type of project it is and also depends on the standards followed by the modeling and designing team.

For this project the data structures we need to implement are stacks, arrays, linked lists, and queues. This way the data can be organized into these different forms and then can be placed into the different pieces of the project. The data that needs to be accessed by multiple components are the growth and starting pollution numbers of the citizens, before it is sorted into the growth of the different buildings and industries. Other shared data would be how many buildings in general were constructed over time in the city. The data can flow through

components by using the different functions previously stated and making them separate files that all the different elements of the project can connect to. Our initial assumptions were that the city would look similar to a grid layout, and that a good approach to organizing our cell data would be to store it in a similar fashion, such as a 2D array. Growth conditions for each tile is different, which means cells within the city need to be able to reference the data stored in their cells to check whether growth conditions are met, so understanding and strengthening the connection between the database and the dynamic map are crucial for functionality.

File Organization:

Plays a vital role in order to separate the logic layer with the presentation layer as file standards should be maintained in the project and the intermediate layer should be there for example controller to control over the presentation layer.

For file organization each file would have specific comments showing what it can do. Organization is a very important aspect of any demanding project, and coding is no different. With the many different file types involved in creating a cohesive, deliverable code, it's crucial that files remain organized and properly separated. At the lowest level we have two main types of files. .cpp and .h. ".h" files contain important code absent from main that serves as the toolkit from which main derives its instructional operations. These files, with specific tasks, should be easily reachable, distinguishable and properly labeled according to their task. Using comments at the beginning of our files can help distinguish related and intertwined files that are dependent on one another.

Accessibility, readability and standardization. File types must remain consistent to avoid errors, and they must use recognizable language. Keeping the files in an accessible bin, directory or folder provides a clear location for project resources/components so that project members can send and receive updates. The ultimate goal is to minimize the time we spend looking through previously completed code, or searching for misplaced project components.