Procedurally Generating Interesting Structures in 3D Video Games

The level design phase in video games represents a critical and labour-intensive process, requiring designers to meticulously place objects manually within a three-dimensionsal environment. Traditionally, this task has been performed manually, often requiring significant time and effort from designers. This study seeks to streamline the level design process by proposing an innovative solution that employs procedural generation techniques, automating the population of game levels with structures in a coherent, interesting and visually appealing manner.

Minecraft was chosen to be the target environment of this project on account of its widespread popularity and its voxel-based graphics allowing expressive manipulation of the scenery. To achieve the desired automation, a custom tool was developed, capable of parsing an input paragraph describing a scene, (e.g., "A tall oak tree near a brick house, on top of a mountain"). The tool interprets this description in order to generate the appropriate geometry within the Minecraft environment. To this end, a Domain Specific Lanugage (DSL) was devised, which allows for the definition of structures in terms of geometric primitives, other structures, and operations applied to and between them. This DSL not only streamlines the process of creating new structures but also provides a robust framework for generating complex and diverse environments.

A dedicated library was then implemented to enable to computation of geometry in terms of the positions and materials of individual voxels. This library enables the efficient generation of intricate structures and landscapes within the Minecraft world, reducing the manual workload for designers while maintaining a high level of quality and detail. In order to integrate the generated structures and landscapes into the game's three-dimensional environment, the Spigot Server application was employed, in conjunction with the MCPI Python Plugin. This combination allowed for seamless communication between the procedural generation tool and the Minecraft environment, ensuring that the generated scenes were accurately represented in the game world.

The system was tested using a variety prompts, simulating real-world use cases and challenging the system's capabilities. The resulting output was evaluated according to a range of objective and subjective criteria, including the fidelity of the generated structures to their descriptions, the visual coherence and appeal of the scene, and the efficiency of the genereation process

The evaluation shows that by automating the creation of immersive and engaging environments, designers can dedicate their focus to refining gameplay elements, optimizing performance, and enhancing the overall player experience. Moreover, the procedural generation techniques employed in the system contribute to a more dynamic and replayable gaming experience, as the generated levels exhibit greater variability and uniqueness.