**Final Project Reflection**

Student: Jime Balvin

Professor: Eugenio Rodriguez

CS - 330 Computer Graphic and Visualization

June 22nd, 2025

When selecting the objects that were going to be part of my scene, I took into account how playgrounds consist of basic geometric shapes (such as boxes, cylinders, cones, pyramids and planes), making it ideal to recreate. Each object served a functional purpose in demonstrating aspects of the course objectives:

* Trees: Cones represent the leaves and cylinders represent the trunks. The use of textures enhanced realism without increasing polygon count.
* Bench: Created from multiple cylinders and boxes. Its development combined challenges like combining rotation, positioning and scaling.
* Trash can: Consisting of a box for the body and an upside down pyramid as the lid; it demonstrated object grouping and rotation.
* Playground structure: This was the most complex part of the scene; it was constructed using several cylinders that act as the support, a pyramid that mimics the roof, boxes that represent platforms and the slide (this is an example of how simple shapes resemble real world items). The repeated objects on this structure (cylinders) act as handrails and steps.
* Sky and ground: These planes were used to define environmental background as well as support for the selected objects.

In order to successfully navigate the 3D scene, the camara allows users to move in all directions and rotate their view naturally:

* W, S, A, D keys respectively allow forward, backward, left and right movement.
* Q and E keys control allows vertical movement (ascending and descending).
* The mouse controls camera orientation. Moving the mouse left or right adjusts the horizontal view, and moving it up or down adjusts the vertical angle.
* The mouse scroll wheel adjusts the speed of the camera movement allowing the user to have finer control.
* P and O keys allow perspective view and orthographic projection.

Code modularity ensures functions were reusable and clearly separate by responsibility. For example, the function that handles transformation logic SetTransformations(), allows modification by adjusting parameters instead of adjusting the logic of the function itself.

The object renderings are based on reusable mesh templates, and the transformations (as mentioned before) are encapsulated making it easy to copy, reuse and adjust.

In conclusion, the project represents a balanced blend of visual creativity, software design principles and technical execution. The playground scene lends itself well to a modular approach and provides a strong foundation for more complex work.