Khulna University of Engineering & Technology,
Khulna-9203, Bangladesh

Project Title: **3D Scene on Street Food Van**

Instructed By:

Dr. Sk. Md. Masudul Ahsan
Professor,
Department of CSE, KUET

Md. Badiuzzaman Shuvo
Lecturer,
Department of CSE, KUET

Submitted By:

Nushrat Tarmin Meem
Roll: **1907083**
Section: B
Year: 4^{th}
Semester: 2nd
Course No: CSE 4208
Date of Submission: 23/01/25

3D Scene on Street Food Van

Introduction:

The project showcases a vibrant 3D scene of a street food van, emphasizing intricate design and attention to realism. The van is detailed with textures, shapes, and lighting to replicate the authentic aesthetics of a bustling food spot. From its structural features to its decorative elements, every aspect is carefully modeled to create a lively and engaging visual experience. The environment surrounding the van is crafted to be visually appealing, with complementary elements like benches, street lights, and trees. By utilizing techniques such as texture mapping and dynamic lighting, the scene transitions seamlessly between day and night modes, enhancing its realism and atmosphere. These additions help to establish a holistic and immersive setting that feels both relatable and imaginative. The project excels in handling diverse object shapes to add variety and complexity. From geometric shapes like hexagons and cones to organic forms like spheres and half-spheres, the integration of these objects adds depth and richness to the scene. This demonstrates the flexibility of 3D modeling and the creative potential of working with varied forms. Beyond aesthetics, the scene serves as a visual prototype for street food businesses, offering a conceptual showcase for entrepreneurs in the food industry. It allows them to envision how a well-designed and attractive food van setup can captivate customers and stand out in competitive markets. Additionally, the project acts as a planning tool for urban designers, helping them visualize and organize street food van setups in public spaces. By providing a realistic representation of how food vans can integrate into an urban environment, it supports decision-making processes related to layout, spacing, and environmental harmony.

Objectives:

- Creation of a realistic 3D street food scene
- Implementation of texture mapping, lighting, curves, fractals, etc.
- Design of diverse object shapes and layouts
- Development of a visual prototype for businesses and planners

Features with Methods:

❑ Different Shaped Objects:

- Hexagon: Constructed using a loop to calculate vertices along a circular path with 6 equal angles. Vertices are connected to form a hexagonal base.
- Cone: Generated by creating a circular base and a single vertex at the apex, connecting the base vertices to the apex to form triangular faces.
- Cylinder: Created using two circular bases (top and bottom) and connecting them with rectangular faces, ensuring the top and bottom circles share the same radius.
- Sphere: Built using spherical coordinate calculations, with stacked circles (latitude) and sectors (longitude) forming the geometry.
- Half-Sphere: A variant of the sphere where only the upper hemisphere is calculated by limiting the latitude angle range.

❑ Lighting (Day and Night):

- Day Lighting: Simulates sunlight by positioning a bright directional light source and using warm ambient and diffuse light to mimic natural daylight.
- Night Lighting: Employs dim, bluish light sources for a nighttime ambiance, with a spotlight or point light to simulate artificial lighting.

❑ Shading (Gouraud and Phong):

- Gouraud Shading: Vertex-based shading where lighting calculations (ambient, diffuse, specular) are performed per vertex, and the colors are interpolated across surfaces.
- Phong Shading: Fragment-based shading where lighting is calculated per fragment (pixel), resulting in smoother and more realistic highlights and shadows.

❑ Texture Mapping:

- Loading textures using libraries like SOIL or stb_image for OpenGL.
- Mapping textures to object surfaces by assigning UV coordinates to vertices. UV coordinates specify how the texture wraps around the geometry.
- Applying the texture using shaders, binding the texture to a sampler in the fragment shader.

Features with Methods:

❑ Bezier Curve Generation:

- Defining control points for the curve in 2D or 3D space.
- Using the **Bezier formula** (de Casteljau's algorithm or recursive blending) to interpolate points along the curve.
- Rendering the curve by generating a line strip using interpolated points between t = 0 and t = 1.

❑ Simple Dynamic Scene:

- Based on axis dx, dy, dz is added to flag bit to control movement for birds and rocket to create simple dynamic scene.

❑ Fractals:

- Generating fractal shapes like the Koch Snowflake or Sierpinski Triangle using recursive algorithms. At each recursion, replacing edges or vertices with smaller, self-similar patterns. These shapes visually demonstrate mathematical recursion and add artistic complexity to the scene.

Result Analysis:

Figure 1: Hexagon Shape

In figure 1, Hexagonal shapes have been created by defining their vertices based on a circle divided into six equal parts. The angles for each vertex have been calculated using the formula:

$$\text{Angle} = 2\pi \times i/6$$

where i has been varied from 0 to 5 to represent the six vertices. The coordinates of the vertices have been determined using the radius of the hexagon and trigonometric functions:

$$X = \text{radius} \times \cos(\text{angle}),\ Y = \text{radius} \times \sin(\text{angle})$$

The calculated vertex data has been stored in an array or buffer, and the hexagon has been constructed by forming six triangles sharing a central point. For 3D shapes, the hexagon has been extruded along the z-axis to form a prism. Shading techniques like Gouraud or Phong have been applied, and textures have been mapped onto the faces to enhance realism and visual appeal. This approach has ensured that the hexagonal shapes are both symmetrical and aesthetically integrated into the scene.

In figure 2, The cone has been created by defining its circular base and connecting it to an apex point. The base has been generated by calculating points along the circumference of a circle using the parametric equations:

$$X = r \cdot \cos(\theta),\ Y = r \cdot \sin(\theta),\ Z = 0$$

where r is the radius of the base, and θ varies from 0 to 2π in equal increments to determine multiple points on the circle. The apex point has been defined at a specific height (0,0,h). To form the conical surface, triangular faces have been constructed by connecting each pair of consecutive points on the base to the apex. The base has been optionally closed by forming a triangular fan using the base points and the center of the circle. Additionally, shading techniques such as Gouraud or Phong shading have been applied to ensure smooth lighting transitions, and texture mapping has been implemented by projecting 2D textures onto the cone's surface, aligning the texture coordinates with the vertices. This process has resulted in a smooth, visually accurate 3D cone seamlessly integrated into the scene.

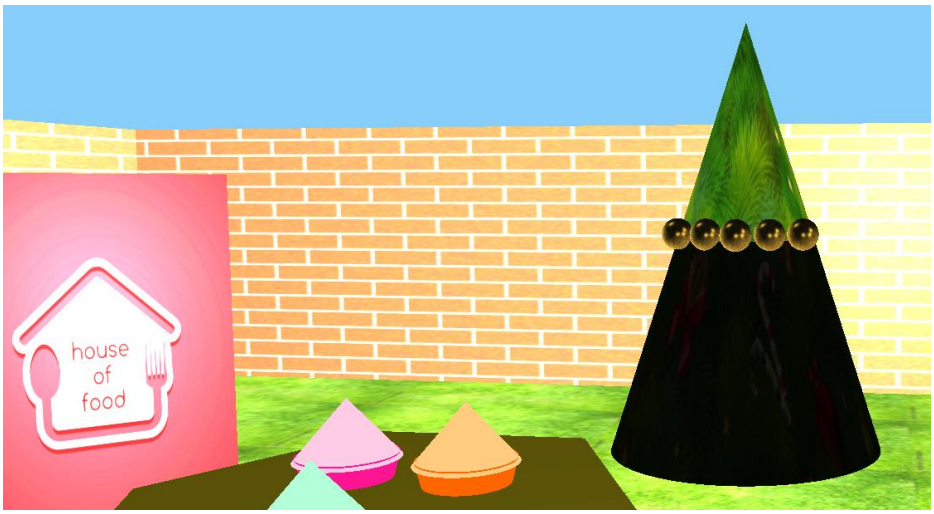

Figure 2: Cone Shape

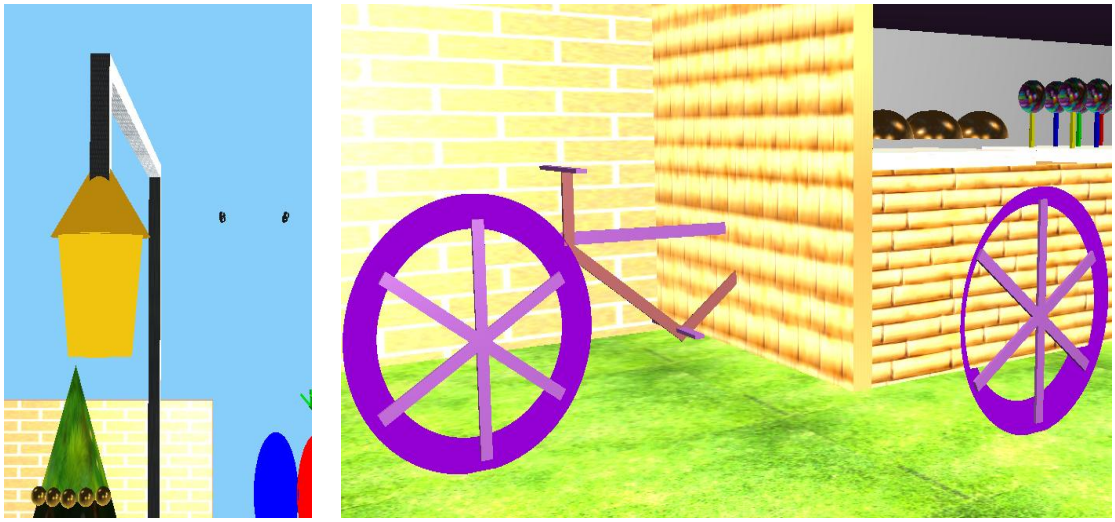

Figure 3: Cylinder for a street lamp and wheels

In figure 3, The cylinder has been created by defining its two circular bases and connecting them with a curved surface. The circular bases have been generated using the parametric equations:

$$X = r \cdot \cos(\theta),\ Y = r \cdot \sin(\theta)$$

where r is the radius of the cylinder, θ varies from 0 to 2π, and z is fixed for each base. One base has been positioned at z=0, and the other at $z=h$, where h is the height of the cylinder. To form the curved surface, vertices corresponding to each point on the top and bottom circles have been connected using quadrilateral faces. Each quadrilateral has been formed by linking two consecutive points on the lower circle to the corresponding two points on the upper circle. The bases of the cylinder have been optionally closed by forming triangular fans, using the center points of the top and bottom circles. Shading techniques, such as Gouraud or Phong shading, have been applied to create smooth lighting effects, and texture mapping has been used to wrap 2D textures around the curved surface, ensuring proper alignment and visual realism. This process has resulted in a complete, smooth 3D cylinder, seamlessly integrated into the scene with accurate geometry and shading effects.

For figure 4, the sphere has been created by defining its surface using spherical coordinates and converting them to Cartesian coordinates. The parametric equations used for generating the sphere are:

$$X = r \cdot \sin(\phi) \cdot \cos(\theta),\ Y = r \cdot \sin(\phi) \cdot \sin(\theta),\ \ Z = r \cdot \cos(\phi)$$

Here, r is the radius of the sphere, ϕ (phi) is the polar angle, varying from 0 to π.θ (theta) is the azimuthal angle, varying from 0 to 2π. The sphere's surface has been divided into small segments by discretizing ϕ and θ into a grid. For each pair of angles, a vertex on the sphere is calculated using the above equations.

To construct the geometry:

Vertices: Vertices have been generated for all combinations of ϕ and θ. Triangles/Quads: The sphere surface has been approximated by connecting adjacent vertices into triangular or quadrilateral faces. These faces form a smooth approximation of the spherical

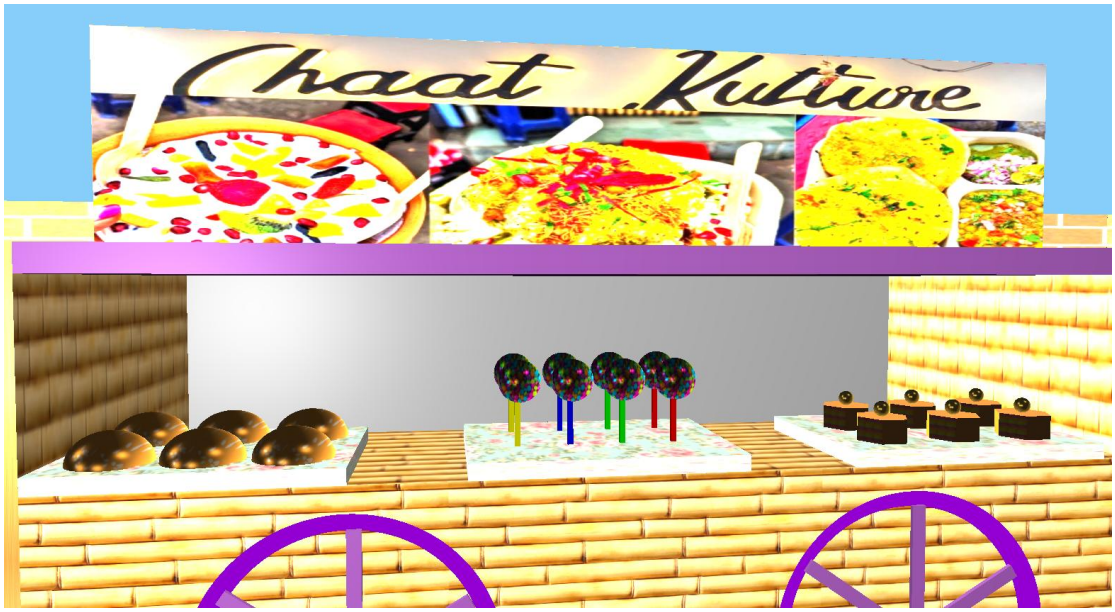

Figure 4: Sphere for candy and top of cakes whereas half-sphere for bread or bun

Surface Normals: Normals for each vertex have been calculated as $n^{\rightarrow} = v^{\rightarrow} \| v^{\rightarrow} \| n = \| v \| v$, where vv is the vertex position, ensuring proper lighting effects. Texture Mapping: The spherical surface has been texture-mapped using polar coordinates. Texture coordinates (u,v) have been derived as:

$$U = 2\pi\theta \, , V = 1 - \pi\phi$$

Shading techniques, such as Gouraud or Phong shading, have been applied to render the sphere with smooth lighting. The result is a visually accurate and smoothly rendered 3D sphere, seamlessly integrated into the scene with texture and light effects.

In the 3D scene, point, spot, and directional lights have been generated using the Phong shading model, combining ambient, diffuse, and specular lighting components for realistic illumination. Point light has been implemented as a source emitting light uniformly in all directions, with its intensity calculated based on the vector from the surface point to the light source. Diffuse and specular terms are derived using the surface normal and view vector, ensuring smooth reflections. Spotlight has been created by defining a light position and direction, applying an attenuation factor based on the angle between the light's direction and the surface point. The spotlight effect focuses the illumination within a cone, with light intensity gradually reducing outside the defined cone angle. Directional light simulates a distant source like the sun, where parallel rays are assumed, and light direction remains constant across the scene.

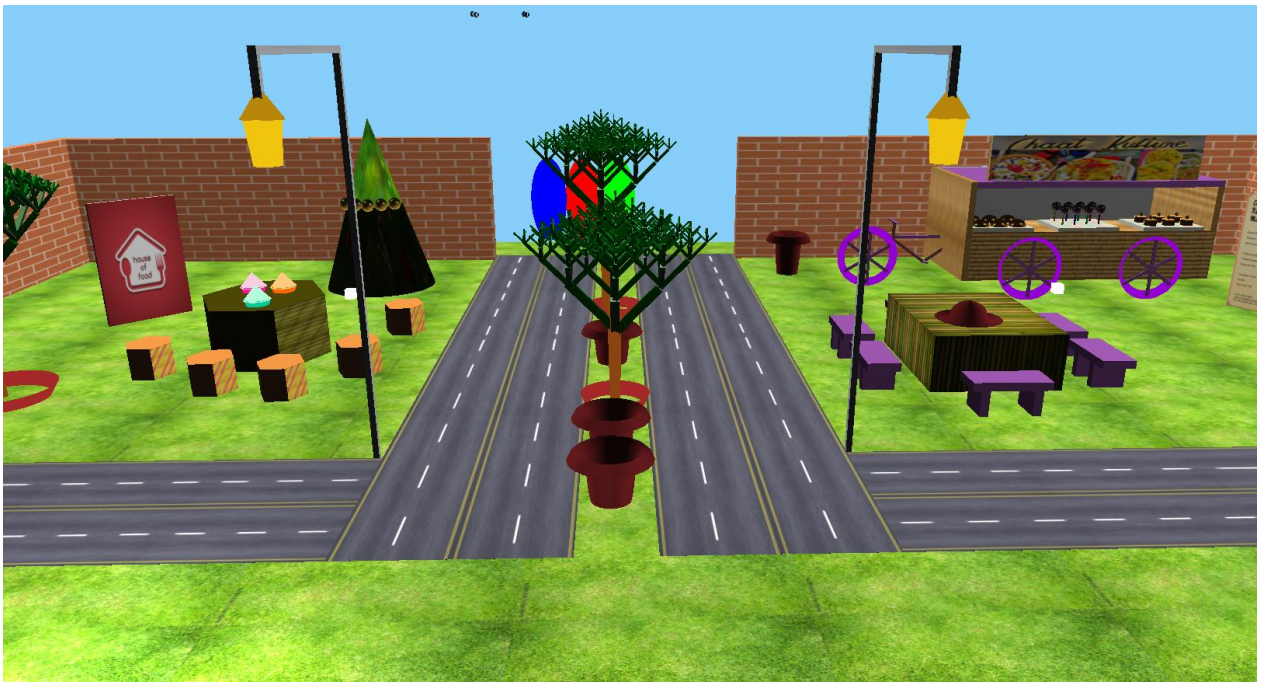

Figure 5: Spot lights 1 and 3 off (one example shown only)

Using Phong shading, lighting calculations are performed per pixel, with interpolated normals passed from the vertex shader to the fragment shader, ensuring smooth and realistic lighting effects throughout the scene.

Figure 6: Spot lights 1 and 3 on (one example shown only)

Bezier curves are generated using a mathematical approach based on control points and Bernstein polynomials. For creating objects like a hat or a vase, the process begins with defining a set of control points, $P0$, $P1$, ..., Pn that outline the desired profile. The curve is mathematically described by the formula:

$$B(t) = \sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-i} t^i P_i$$

Here, B(t) is the point on the curve corresponding to the parameter t, which varies from 0 to 1. (ni) is the binomial coefficient, and Pi is the control point. By sampling tt at regular intervals, a sequence of vertices is generated, creating the curve's profile. This 2D curve is the foundation for constructing 3D objects. To create a 3D hat or vase, the Bezier curve is rotated around an axis, typically the y-axis, forming a surface of revolution. For each vertex (x,y,z) on the curve, new positions are calculated during rotation using the formulas:

$$x' = x \cdot \cos(\theta) - z \cdot \sin(\theta)$$

$$z' = x \cdot \sin(\theta) + z \cdot \cos(\theta)$$

Here, θ represents the angle of rotation, which is incremented from 0 to 2π. The vertices generated at each step are connected using indices to form triangles or quads, creating a smooth mesh for the object. For a vase, the control points define a curved profile that narrows and widens, while for a hat, the control points define the brim and crown. This method ensures precision and smoothness in the design, making Bezier curves a powerful tool for complex shapes.

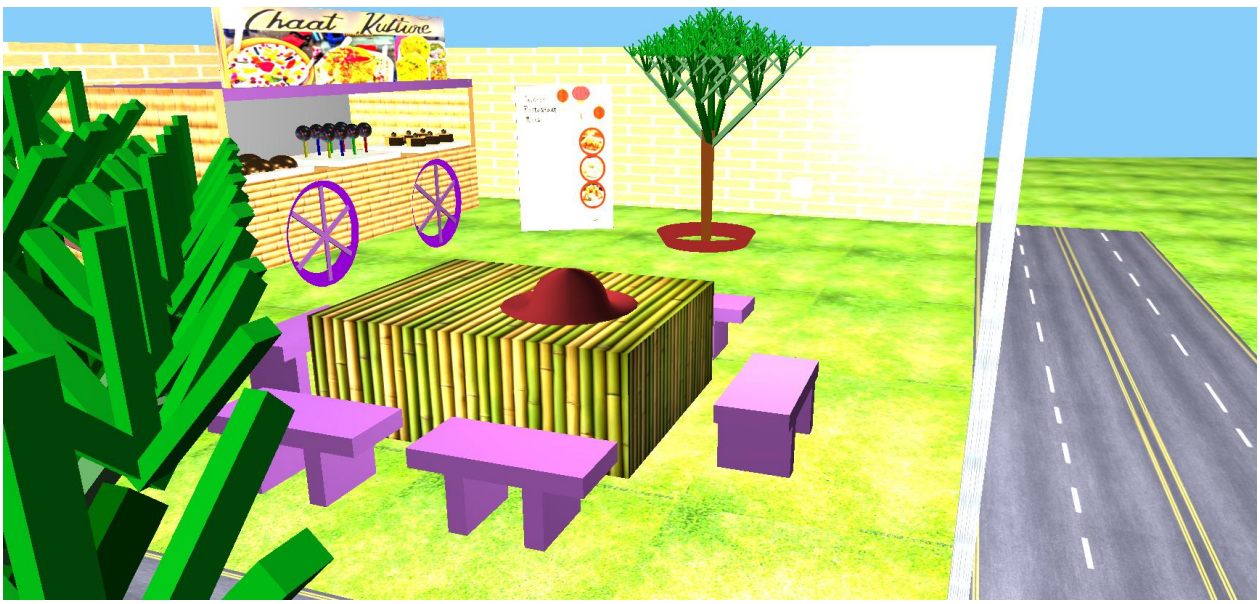

Figure 7: Curvy object hat and fractal trees

Fractals are generated to create a tree by using recursive algorithms that mimic the natural branching structure of trees. Starting with a trunk represented by a simple line or cylinder, the tree's branches are recursively divided into smaller branches, each with reduced size and rotated at an angle. The angle of branching and the length reduction factor are key to simulating the natural look of tree growth. The recursion formula involves adjusting the position of each branch using the cosine and sine functions for horizontal and vertical placement, while the length of each branch decreases with each recursive step. This self-similar structure creates a fractal tree, where each branch pattern is repeated at smaller scales, resulting in the intricate, natural appearance of a tree.

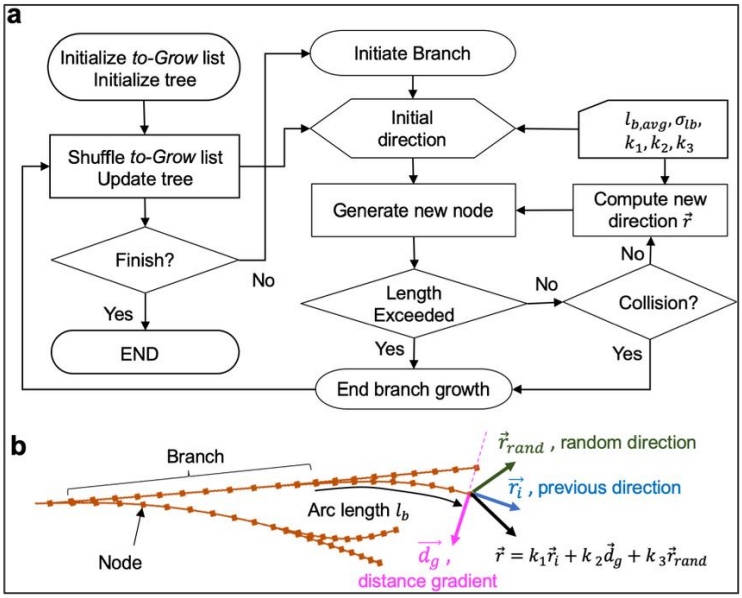

Figure 8: Fractal trees generation flow chart

Figure 9: Curvy object vase and fractal trees

The rocket's upward movement was achieved by translating its 3D model along the y-axis. A continuous upward translation was applied at a constant speed, determined by the frame rate, to simulate the motion. Keyframe animation was used to control the rocket’s position at specific intervals, ensuring a smooth transition between frames. Gravity was not simulated in this animation, and the movement was kept linear, focusing solely on the illusion of upward travel. The animation was rendered frame by frame, creating a smooth upward motion for the rocket.

The bird's movement was designed to follow a random trajectory in 3D space along the x, y, and z axes. A random offset was applied to its position in each frame, allowing for unpredictable changes in its flight path. The bird's motion was not confined to a specific direction, but instead moved freely along all three axes, creating a more natural and dynamic flight behavior. The speed of movement was randomized within a set range, ensuring variations in how fast or slow the bird moved at different times. The random movement was applied using a procedural approach, adjusting the bird's position by small increments at each frame, resulting in an organic, erratic flight pattern.
To simulate realistic bird behavior, slight pauses and direction changes were incorporated, ensuring that the bird’s movement was not entirely chaotic. This allowed for a more lifelike animation, where the bird's position was adjusted in small random steps on the x, y, and z axes in each frame. The final rendering displayed the bird’s erratic flight, with continuous updates to its position, creating the illusion of free and random movement in three-dimensional space.

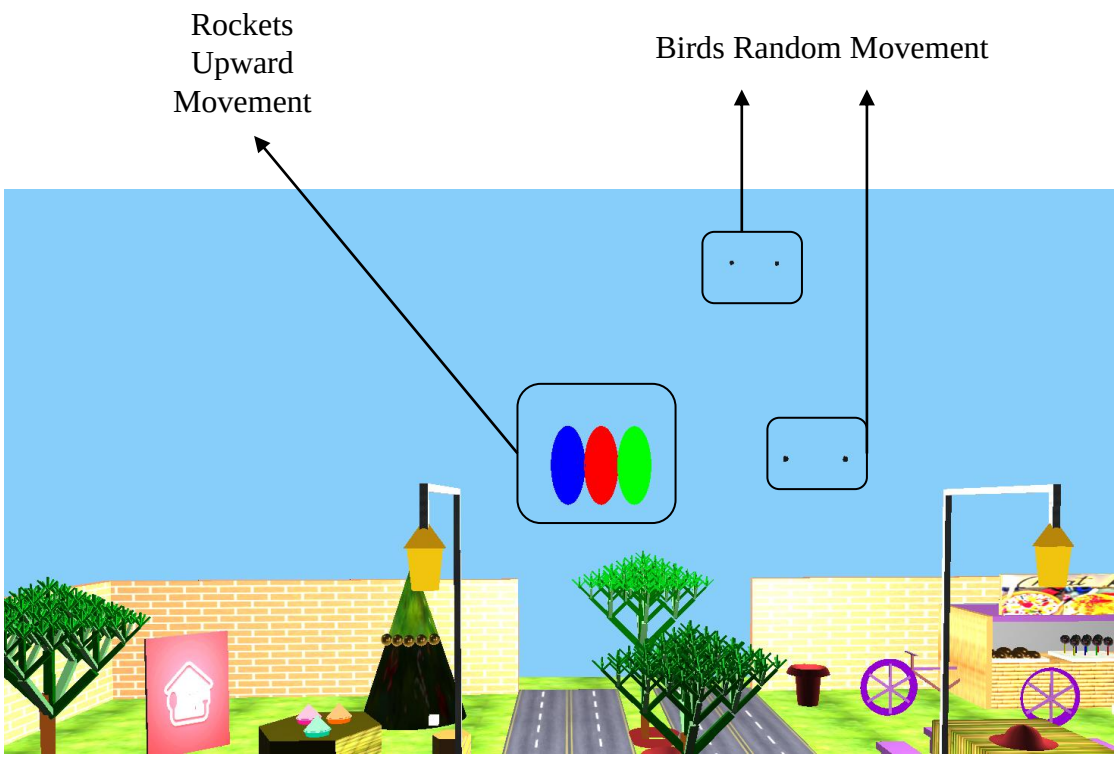

Figure 10: Very Simple Animation

Figure 11: Night Mode On (Key P)

In daylight mode, the scene is illuminated with bright, natural lighting, simulating a vibrant daytime environment where the sun casts strong shadows, highlighting the details of the food van and surrounding objects. The warm light enhances the colors and textures, creating a lively atmosphere with a bustling street scene. In contrast, night mode introduces a more subdued, atmospheric ambiance, with artificial lighting from street lamps and neon signs providing the primary illumination. The van is bathed in warm artificial lights, contrasting with the cooler, darker surroundings, creating a cozy and intimate setting. The softened shadows and deep blue sky evoke a sense of mystery, transforming the scene into a welcoming, late-night street food experience.

Figure 12: Main Food Van

Main Key Functionalities:

- ❑ W – Forward
- ❑ S – Backward
- ❑ A – Left
- ❑ D – Right
- ❑ R – Up
- ❑ X – Down
- ❑ O – Day mode
- ❑ P – Night mode
- ❑ 3, 4, 7, 8, 9, 0 – Lights Control
- ❑ X, Y, Z – Rotation
- ❑ H, I, J - Scaling
- ❑ Q – Camera
- ❑ M – Animation
- ❑ F – Birds Eye
- ❑ On mouse click functionality
- ❑ Scroll Bar or Touch Pad for all together

Conclusion:

In conclusion, the 3D street food van project exemplifies a successful fusion of technical skills and creative innovation, resulting in a visually engaging and highly detailed scene. The incorporation of complex design elements, such as dynamic lighting, varied object shapes, and meticulous texture mapping, contributes to the development of a vibrant and immersive environment. The scene's seamless transition between day and night further enhances its realism, creating a dynamic atmosphere that reflects the real-world changes of a public space. This transition not only elevates the visual appeal but also adds an element of realism, making the scene feel alive and ever-changing. Beyond its aesthetic appeal, the project holds practical value as a conceptual prototype for street food businesses. It offers a visual representation that can aid in planning and layout design, providing valuable insights into the spatial arrangement and functionality of such businesses in urban spaces. Urban designers can utilize this model to visualize how a street food setup might integrate into various cityscapes, taking into account both functionality and artistic appeal. Overall, this project demonstrates the immense potential of 3D modeling in creating not only aesthetically pleasing visuals but also meaningful, functional representations that bridge the gap between creativity and practical application. It highlights the versatility of 3D modeling as a powerful tool for transforming abstract ideas into tangible, immersive experiences.