CAP-4710/5701 Computer Graphics

Assignment 4

FIU Knight Foundation School of Comp. & Info. Sciences

1 Exercises (30 points)

1. (10 points) Write a pseudo-code to set up the matrix that transforms world-coordinate positions to 3D viewing coordinates, given P_0, N, and V. The view-up vector can be in any direction that is not parallel to N.

2. (20 points) Set up an algorithm for converting a given torus with an ellipsoidal cross section to a polygon mesh representation.

2 Programming Assignment (70 points)

For this programming assignment, you need to implement the first phase of the game "3D Snakes" program[1]. In the following programming assignments, you will implement the next phases of the game to complete its implementation.

2.1 First Draft

In the program's first draft posted on Canvas, the polygon mesh representation of a snake body can be drawn using the basic quadratic surfaces: cylinder and torus. Figure 1 shows the polygon mesh drawn by the initial draft of the program.

In this figure, you can see a total of 15 right circular cylinders connected by 14 quarter tori in a sequential manner. The axes of every two consecutive cylinders in this figure are perpendicular. Every cylinder is considered to be a segment of the snake body in the snake game and every quarter torus specifies the point in which the body of snake turns to a different direction.

In the given program, the snake body is drawn using the following function that has no return value, but gets two input parameters: a double value representing the radius of cylinders modeling the snake body and a variable of type enum ColorPattern that can either be CHECKERED (default), SOLID, V_STRIPPED, and H_STRIPPED.

```
void snake(double r, ColorPattern color);
```

[1]The 2D version of this game can be played here: https://www.google.com/search?q=play+snake

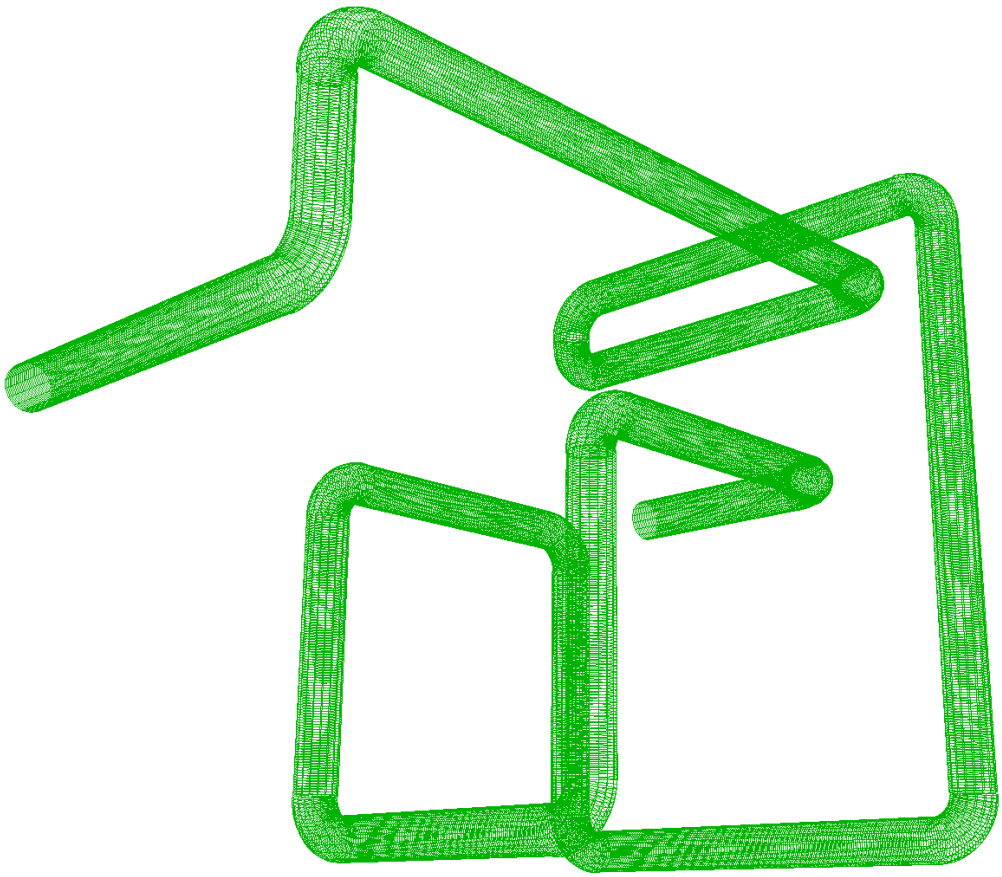

Figure 1: A polygon mesh used for boundary representation of the body of a snake in three dimensions. The snake has no head or tail!

Also, the coordinates of the snake body are stored and retrieved using an external array of 3D points declared by Point snakeBody[MAX_SEGMENTS]; where Point is a structure containing three doubles for Cartesian coordinates x, y, z, and MAX_SEGMENTS is a constant equal to 1000. For example, the snake body drawn in Figure 1 is stored in the snakeBody array in the following way:

```
snakeBody[0] = { -0.1,0,0 }; snakeBody[1] = {-0.5, 0, 0 };
snakeBody[2] = { -0.5,0, 0.5 }; snakeBody[3] = { -0.5,0.5, 0.5 };
snakeBody[4] = { 0,0.5, 0.5 }; snakeBody[5] = { 0,0, 0.5 };
snakeBody[6] = { 0,0, 0 }; snakeBody[7] = { 0,0.5, 0 };
snakeBody[8] = { -0.7,0.5, 0 }; snakeBody[9] = { -0.7,-0.4, 0 };
snakeBody[10] = { 0,-0.4, 0 }; snakeBody[11] = { 0,-0.27, 0 };
snakeBody[12] = { -0.6,-.27, 0 }; snakeBody[13] = { -0.6,-.27, 1 };
snakeBody[14] = { -0.6,0, 1 }; snakeBody[15] = { 0,0, 1 };
```

In this example, the points stored in any two consecutive cells of the array specify one segment of the snake body represented by a cylinder in Figure 1, The first and last points specify where the tail and head must be placed.

2.2 (20 points) What Needs to be done to the First Draft

You should change the program that allows user to use keyboard to interact with the program. To zoom-in/zoom-out, you can press 'L' and 'S' respectively. To rotate the scene about x, y, or z axis, you can press 'X', 'Y', or 'Z' respectively.

2.3 Second Draft

In the program's second draft posted on Canvas, the scene is made of a snake (with a specific location), a number of bunnies (scattered randomly over a $2 \times 2 \times 2$ cube), and a cube specifying the boundaries of the game field. Figure 1 shows the first scene of the game provided by the initial draft of the program.

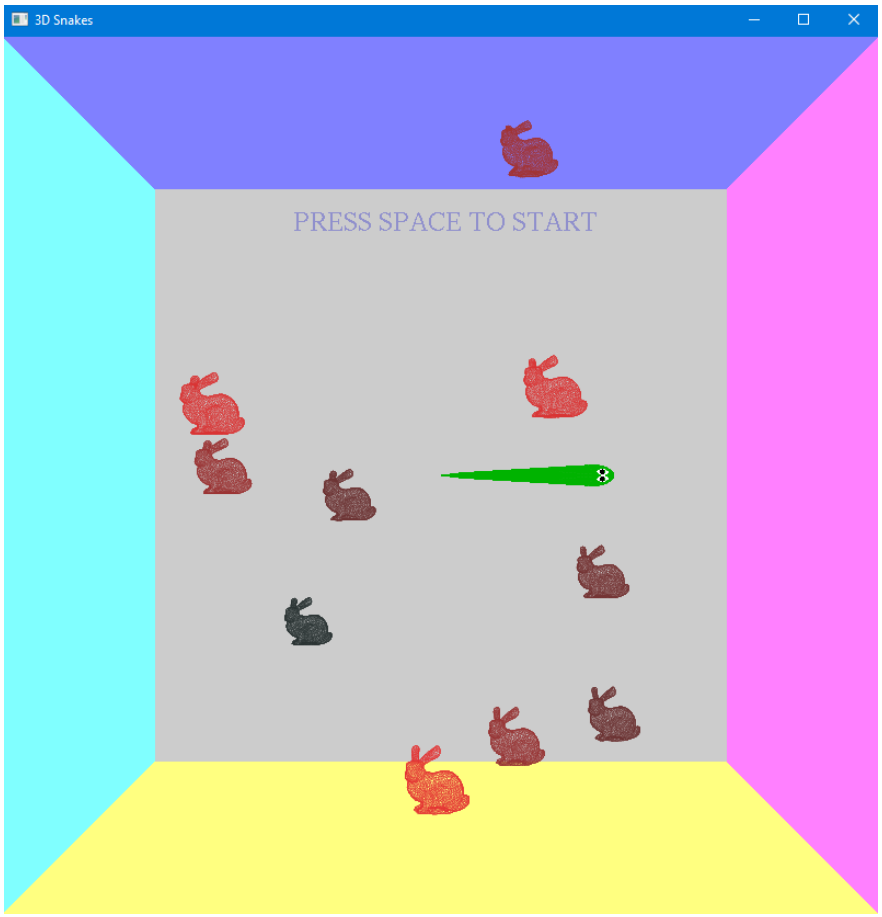

Figure 2: Starting scene of the game.

You can press space bar to make snake start moving. The snake moves in the direction that its head and eyes are pointing to. In the case that the user presses one of the four arrow keys, the snake turns to right, left, up, or down (See Figure 2). By pressing space bar again, you can pause the game.

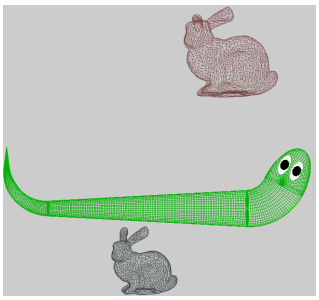

Figure 3: The way snake reacts when pressing arrow keys (up-arrow in this case)

2.4 (50 points) What Needs to be done to the Second Draft

1. (25 points) Change the program so that any moment in time, the user can use one of the following keys:

- x: to rotate the scene about the x-axis
- X: to rotate the scene about the x-axis (opposite direction)
- y: to rotate the scene about the y-axis
- Y: to rotate the scene about the y-axis (opposite direction)
- z: to rotate the scene about the z-axis
- Z: to rotate the scene about the z-axis (opposite direction)
- Home: to pan the camera upward
- End: to pan the camera downward
- PgUp: to pan the camera rightward
- Insert: to pan the camera leftward
- +/-: to zoom in/out

2. (25 points) Let the user decide on where the camera should be and whether it has to move by the snake. In the initial draft, the camera is positioned on the z-axis and looks down at the scene. You need to provide a second option for the user to make the camera positioned on the snake eyes and displays the scene that snake sees. When snake moves, camera moves as well. The user must be able to switch between the static (first) and dynamic (second) options by pressing keys '1' and '2'.

3 Deliverables

Submit a zip file containing the following items:

- A pdf file containing your answer to the exercises.
- A readme.txt file containing the names, panther ids, and email addresses of students working on the assignment if you have done the assignment as a group of two.
- The source/resource files of the updated first draft.
- The source/resource files of the updated second draft.
- A short mp4 video (less than 4 minutes) interacting with the completed first draft and the completed second draft.