Snake Dimensions - Documentation

Daniel Barovbe

January 15, 2018

Contents

1 Running the Game

To compile the game, type ***make snakeD*** in the terminal in the game's directory.
To run the game, type ***./snakeD*** or click on the executable file in the file explorer.

2 Menu

When launching the game, the main screen in Figure 1 appears, giving the controls at the bottom of the window and the highest score being displayed at the top. The world is displayed to the user, rotating to allow the user to see all the obstacles before entering the world whilst showcasing its design.

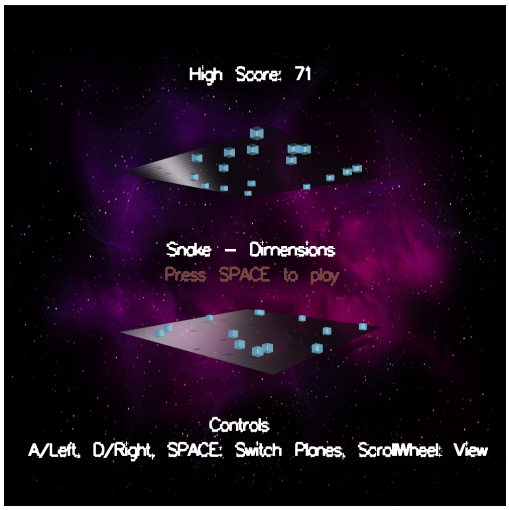

Figure 1: The Menu Screen

3 Controls

When the user presses the **SPACE** key, they are thrown into the three-dimensional world as the snake. To navigate the world the **LEFT/a** and **RIGHT/d** keys are pressed to turn left and right relative to the snake. The camera follows the snake from behind or above depending on the selected view, and rotates to remain focused on the snake's head. To change the view, the **SCROLL WHEEL** is scrolled up to have an immersive 3rd person persective of the world, whilst scrolling down gives a more top-down perspective for a better view of the entire plane.

Figure 2: The multiple camera perspectives selectable during gameplay time

As shown in Figure 1, the world consists of two planes. Press the **SPACE** key in order to switch between the planes. The snake will ascend or descend depending on what plane it is currently on.

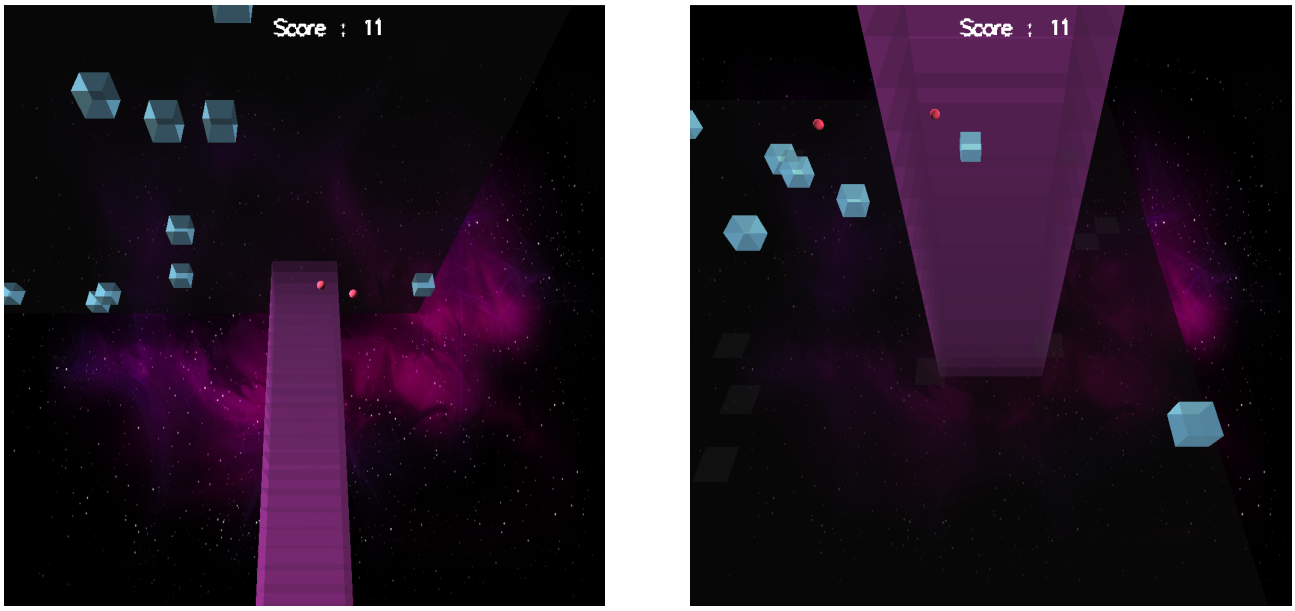

Figure 3: The snake switching planes once the SPACE key is pressed

4 Gameplay

The gameplay on a single plane has all the basic elements of the original snake game but with a few additional features. As shown in Figures 2 and 3, there are various objects in the world that the player must be weary of. The blue translucent boxes are randomly generated **obstacles** which must be avoided. Colliding with these will cause the player to lose and go back to the main screen. They grey squares on the ground are the **"shadows"** of the **obstacles** above/below, and switching planes whilst on these squares will cause the snake to hit the obstacle in the opposite plane. The red spheres are **food**, and eating them will cause the snake to grow and the score to increment by 1. The snake moves faster as the score increases. Finally, when the snake reaches the end of the world it will wrap around the screen and emerge from the opposite side.

5 Design

The 3D perspective camera is transformed to follow the snake's head as the reference point. To change the perspective, the y reference point of the camera is altered. The camera movements are interpolated for smoothing effects using the following formula:

$$\mathbf{C} = \mathbf{C} + \alpha(\mathbf{X} - \mathbf{C}), \quad \alpha \leq 1 \tag{1}$$

Where $\mathbf{C}$ is the camera's position vector, $\mathbf{X}$ is the camera's final position vector and α is the interpolation factor which affects the speed.

The movement system of the snake is continuous rather than discrete which allows for fluid motions and smoother gameplay. A texture is drawn on a single quad the size of the screen used as the background with **glBindTexture()**, and anti-aliasing is used to smooth the edges of the text using **glEnable(GL_LINE_SMOOTH)**. Four diffuse and specular lights are used, located far away from each level corner. The level flooring uses a combination of alpha blending and specular materials with **glEnable(GL_BLEND)** and **glMaterialfv()** for the specular component to give a reflective glass effect as in Figure 1, and consists of many quads which each reflect light differently in the world. The obstacles in the world use blending to allow the player to see other objects behind the obstacles when in 3rd

person view. Finally, the segments of the snake between the planes are made translucent with blending and the depth buffer set to read only with **glDepthMask(GL_FALSE)** to prevent it from obstructing the players view of the plane whilst switching.

Word Count : 692