ABSTRACT

In the field of computer science, researchers have done much work involving the animation of three dimensional characters. Some of the more important animation related subjects for study include behavior planning, animation of flocks, and artificial life. This paper seeks to discuss such issues by comparing and contrasting different approaches to three dimensional animation. We will also provide a demonstration of the various techniques through a simplified two dimensional animated program with which the user may interact.

Three dimensional computer animation has many applications, including entertainment, industry, and education. Designing three dimensional animations that are both realistic and manageable is no small task. Clever algorithms are required to animate characters in a realistic and efficient manner. Particularly tricky are animations involving flocks of multiple characters which must interact with one another to determine their future velocities and actions. To avoid higher computational overhead, flocking algorithms generally use methods that check a character’s velocity and position only against nearby characters, and then adjust that velocity and position to adjust to match those neighbors. (Anderson, Chenney, McDaniel, 2003). Researchers have developed many techniques for streamlining and simplifying the animation process. For instance, animations may be strung together from simpler animations based on a character’s surroundings (Lee, Lee, Myung, 2006).

To help foster an understanding of three dimensional animation, we designed a two dimensional animated program featuring sprites that mimic the animation techniques seen in more sophisticated three dimensional programs. This program can be recreated using most standard compilers (in this case, Visual Basic.net) with GUI capabilities and the proper graphics packages, without installing expensive software or graphics cards.

Models are often useful for describing more complicated phenomena. For example, an architect could use a computer simulation to test the structural integrity of a building before it is built. A hobbyist might build model cars or airplanes to better understand the genuine article. Similarly, a two-dimensional program can be used to demonstrate techniques that can be applied to more complicated three-dimensional animation.

This program consists of a nine by six field of squares. Each of these squares contains an image of one of several types of terrain. This image may be changed to one of an array of eleven images by clicking on the square. The terrain images consist of a flat stony path, various stone walls in all directions, an iced-over surface, and a square containing an oversized can of tuna. One or more animated sprites appear on this nine by six grid. This animated character reacts to the terrain based on a two dimensional array, which records the type of terrain that each square holds. Also, a red ball appears somewhere on the grid. Below the grid, there is a drop-down menu where the user may select one of four characters to appear on the grid. These choices are “Ghost,” “Penguin,” “Rabbit,” and “Multiple Sprites.”

The characters are drawn as sprites (distinct images with a transparent background color) on the two-dimensional playing field. Lime-green is set as a transparent color. In other words, when a picture is drawn, the lime-green is filtered out: The graphics functions will not draw the lime-green color, so the background will be completely visible through any part of the sprite which was lime-green in the image file. A timer in the Windows form ticks at a constant rate, and the characters move to their next animation frame and screen position on each tick. Each of these animated characters exhibits its own unique characteristics, and will react differently to the changing environment. They each have their own “move” method. This move method determines how the character will react to the terrain, the red ball, and the boundaries of the field.
REFERENCES


