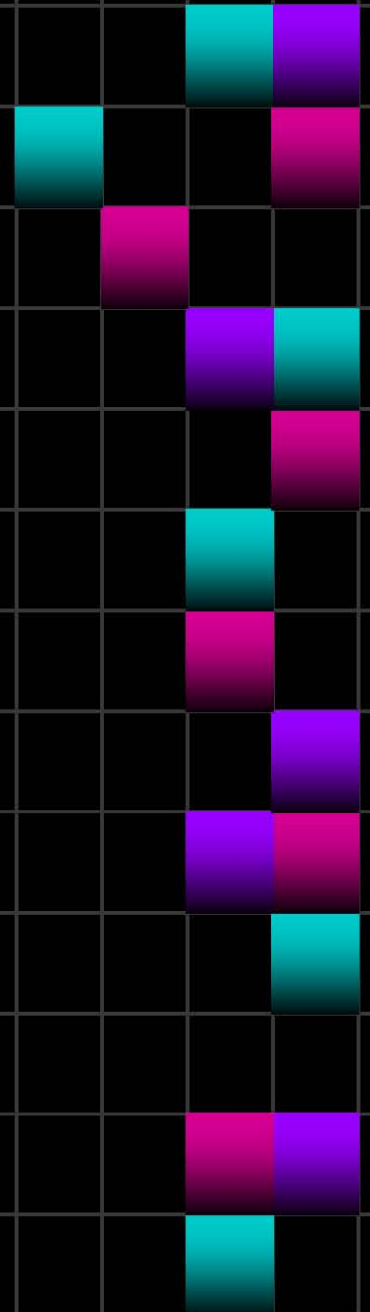


Graphics Animation Using Borland's bgi

Mr. Dave Clausen
La Cañada High School



Graphics Commands for the bgi

- Remember, a summary of Borland's graphics commands for Turbo C++ 3.0 for DOS can be found in a previous lecture.

Animation basic concepts.

- The basic concept for graphics animation is to have some form of loop (definite or indefinite), where we **draw an image**, **erase it** (by drawing it exactly again, only in the background color instead of the foreground color), and **update its coordinates**.
- This process repeats until the image reaches a desired location, or until some other condition is met.

Blinking Objects

- The algorithm for a blinking object is as follows:
 - Draw the object (in a foreground color)
 - For a definite number of iterations do (or while)
 - Erase the object (draw in the background color)
 - Pause for a specified time (as necessary)
 - Draw the object (in a foreground color)

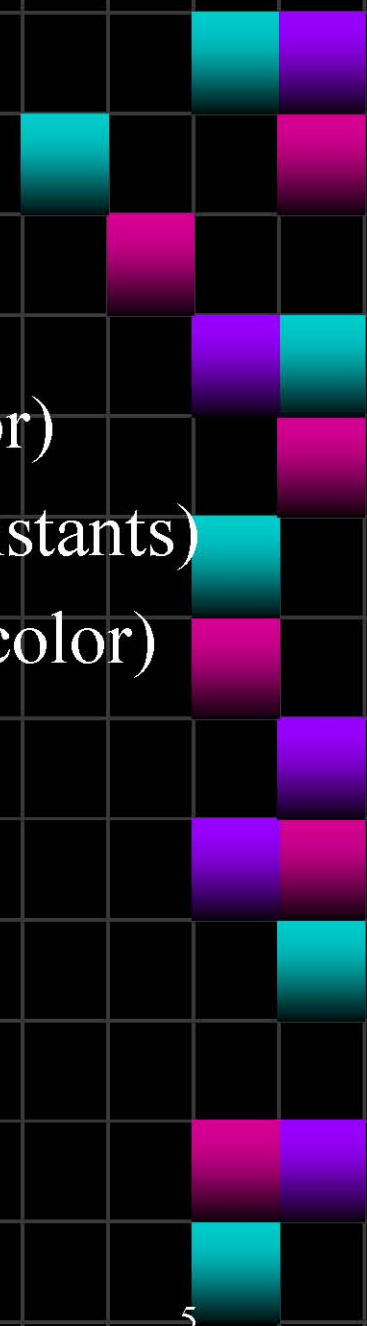
[blink.cpp](#)

blink.txt

[blink.exe](#)

Linear motion

- The algorithm for linear motion is:
 - Draw the image in the foreground color(s)
 - While the image is not at its destination (or for)
 - Pause for a specified time (use symbolic constants)
 - Erase the image (draw it in the background color)
 - Update the image's position
 - Draw the image in the foreground color(s)
 - To update the image's position, increment or decrement:
 - $y = y + \text{delta_y}$; or $y = y - \text{delta_y}$;
 - $x = x + \text{delta_x}$; or $x = x - \text{delta_x}$;



Linear motion: y in terms of x

- The algorithm remains the same as defined on previous slides. The difference is how we calculate the variable Δy in terms of x .
- For linear paths we may use the increment:

$$y = y_1 + (3 * x - 5);$$

//The equation of a line between 2 points is:

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1} (x - x_1)$$

Non linear motion

- The algorithm remains the same as defined on previous slides. The difference is how we calculate the variable `delta_y` in terms of `x`.
- For parabolic paths we may use the function:

$$y = y + (0.01 * (x * x));$$

//The basic formula for a parabola in standard form:

$$y = a(x - h)^2 + k$$

To Move an Object in a Parabolic Path:

- Let's look at this example which doesn't merely draw a parabola, but moves an object in a parabolic path.

Other Examples:

- Let's look at examples of previous student's work.
- They fall into 2 categories:
 - non-interactive animation
 - as previously discussed
 - interactive animation
 - using indefinite loops while not `kbhit()`.
 - Keys can be detected using `getch()`
 - Using `switch` or nested `if`'s to determine action of key hit. (use ASCII codes of the keys.)