Bouncing Ball Part I

(Gravity)

Prepared by: Teton Multimedia (www.TetonMultimedia.com)

Hold a ball in the air, release it, and it falls toward the ground. Question: “How can this be represented mathematically, and how can we use this math within a computer program?” In the following discussion we’ll answer these questions simply and directly.[1]

When the ball falls toward the ground, it accelerates due to gravity. This acceleration is constant near the earth’s surface if air resistance is ignored. Acceleration (a) is the change in speed (velocity) during a specific time-span. Similarly, velocity (v) is the change in location (y) during a specific time-span.

On a highway, velocity is referred to in terms of “miles per hour.” In this reference, the time-span is an hour, and the change in location is a number of miles. Mathematically, we can refer to a time-span as “delta t”, or Δt. Similarly, change in location becomes Δy.

From a mathematical perspective, the “per” in the phrase “miles per hour” means “divided by”. This means that

$$\boxed{60 \textit{ miles per hour} \;=\; (60 \text{ miles}) / (\text{one hour})} \tag{1.1}$$

In this context, Δy is 60 miles and Δt is one hour. If, for example, Δy was 240 miles and Δt was 2 hours, then the speed would be

$$\boxed{240 \textit{ miles} / 2 \textit{ hours} \;=\; 120 \textit{ miles per hour}} \tag{1.2}$$

Another common reference for speed is “feet per second.” Notice that this phrase is similar to “miles per hour” in that it contains a length (feet) and a time span (a second), separated by the word “per.”

Recasting equation 1.1 using symbols, we have a general equation for velocity:

$$\boxed{v \;=\; \Delta y / \Delta t} \tag{1.3}$$

As noted above, acceleration is the change in velocity over a time span, or

$$\boxed{a \;=\; \Delta v / \Delta t} \tag{1.4}$$

Note that equations 1.3 and 1.4 have exactly the same form. This should give you the comfortable feeling that if you can understand velocity, then the concept of acceleration shouldn’t be a problem. Acceleration is the change in velocity over a time-span. For example, the acceleration might by “10 feet per second per second.” This might be rewritten as “(10 feet per second) per second”, where the parenthetical portion is the velocity. This means that after 3 seconds, the velocity will be 30 feet per second (30 ft/sec), and after 7 seconds it will be 70 feet per second (70 ft/sec).

From a mathematical perspective, the phrase “(10 feet per second) per second” can be rewritten as “(10 ft/sec)/sec”, which can be rewritten as “10 ft/sec^2”.

At the instant you release the ball, its y velocity is actually zero. However, its acceleration is *not* zero. On the earth’s surface, its acceleration is 32 ft/sec^2. This means that its velocity will increase. Multiplying both sides of equation 1.4 by Δt, we arrive at

$$\boxed{a \ast \Delta t = \Delta v} \tag{1.5a}$$

or

$$\boxed{\Delta v = a \ast \Delta t} \tag{1.5b}$$

In this equation we’re free to use any Δt we please. However, to represent a moving ball smoothly on a computer display, we have to use a suitably small Δt. As it turns out, an appropriate value is easily determined from motion pictures, which typically run in the vicinity of 30 frames per second. This implies that each frame is separated by 1/30th of a second, and hence, a suitable value of Δt is 0.033 second.

At this point it should be noted that, in this context, Δt is typically referred to as “the time-step size”. At the beginning of the first time-step, the ball’s velocity was zero. At the end of the time-step, equation 1.5b gives us a velocity of

$$\boxed{v = 32 \ast 0.033 = 1.056 \text{ ft/sec}} \tag{1.6}$$

This implies that the *average* velocity during the time-step was

$$\boxed{v_{avg} = (0.0 + 1.056)/2 = 0.528 \text{ ft/sec}} \tag{1.7}$$

Rewriting equation 1.3 by multiplying both side by Δt gives us

$$\boxed{\Delta y = v_{avg} \ast \Delta t} \tag{1.8}$$

Completing the multiplication gives us $y = 0.017$ ft. This is the distance that the ball moves during the first time-step after it is dropped. The math is similar for the second time-step and each time-step thereafter. The velocity at the end of a time-step can be generalized as

$$\boxed{v_e = v_i + \Delta v} \tag{1.9a}$$

and substituting using equation 1.5b gives us

$$\boxed{v_e = v_i + (a \ast \Delta t)} \tag{1.9b}$$

where v_e and v_i are the velocity at the end and beginning of the time-step, respectively. With this information, equation 1.8 can then be generalized as

$$\boxed{y_e = y_i + (v_{avg} \ast \Delta t)} \tag{1.10a}$$

or

$$\boxed{y_e = y_i + ((v_i + v_e) \ast 0.5) \ast \Delta t} \tag{1.10b}$$

Lingo!

The equation set above is sufficient to program a lingo behavior as follows. We assume that the FrameTempo is defined in the tempo channel (Director doesn't let us use lingo specify the FrameTempo). Please not that it *is* possible to further compact both the equations above and the code below; however, the form presented here is compelling from an instructional perspective. Note also that ***the leading line numbers should be omitted from actual code***. They are included here only for clarity.

```
1. Property vi, yi, a, dt, MyS
2.
3. On BeginSprite me
4.    MyS = Sprite(me.SpriteNum)
5.    vi = 0.0
6.    yi = float(MyS.LocV)    -- use a variable to store the location
7.    a = 32.0 * MyS.Height -- assume ball height is 1 foot
9.    dt = 1.0 / the FrameTempo
10. end
11.
12. On ExitFrame me
13.    ve = vi + a*dt         -- equation 1.9b
14.    ye = yi + (ve + vi)*0.5*dt   -- equation 1.10b
15.    MyS.LocV = ye
16.    vi = ve                 -- prepare for next time-step
17.    yi = ye                -- prepare for next time-step
18.    go to the frame
19. end
```

Using this programming in a sprite's behavior will cause the sprite to accelerate and vanish off the bottom of the screen. Note that the sprite's vertical location is calculated and stored in the variable “yi”. The sprite’s LocV property then “grabs” this value. This technique prevents “jerky” the behavior associated with integer round-off that occurs if the calculation uses LocV directly (an integer, by definition). (Note that this type of round-off-related problem is most noticeable at low velocities.)

The “go to the frame” instruction (line 18) simply loops the playback head, causing the ExitFrame script to loop indefinitely. Meanwhile, lines 16 and 17 prepare the loop for the next time-step: the initial values for one time-step are the end value from the preceding time-step.

The gravitational acceleration is defined in Line 7. If it was simply defined as 32.0, then it would be implied that one pixel is a foot high. For a more realistic simulation, we multiply by the height of the ball, implying that the ball is one foot high.

---

1. Puritans may cringe at the lack of mathematical rigor in our presentation. However, it is our feeling that such rigor often clouds what may otherwise be a clear and concise learning experience. We prefer to present concepts accompanied by “rough” mathematics, followed by specific examples, and finally followed by rigorous mumbo-jumbo (if so inclined).