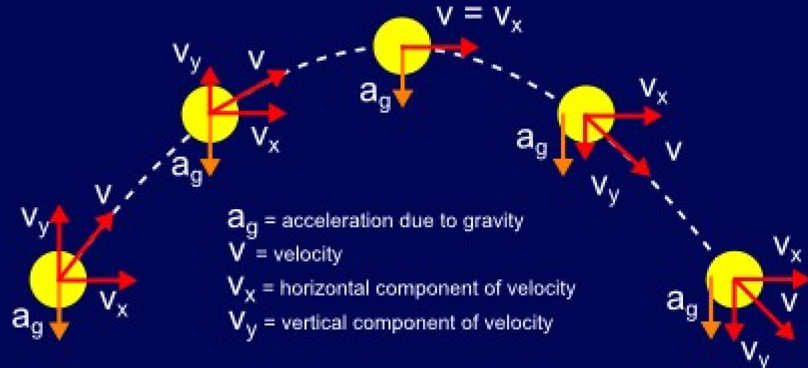
Physics Help

Projectile Motion Background

Projectile motion involves the movement of an object, called a projectile, after it has been launched or thrown into the air. Between launch and landing the only forces acting on the object are gravity and air resistance. However, for this game you can entirely ignore air resistance. One very important property of projectiles is that their horizontal movement (v_x, parallel to the ground) is entirely independent of their vertical motion (v_y, perpendicular to the ground). This concept forms the basis for many of the calculations. Because gravity only acts in the vertical direction and due to Newton's 2nd law, the projectile's horizontal speed remains constant throughout its flight if there's no air resistance. However, the vertical motion of the ball can be modeled using constant linear acceleration. This concept is illustrated below.



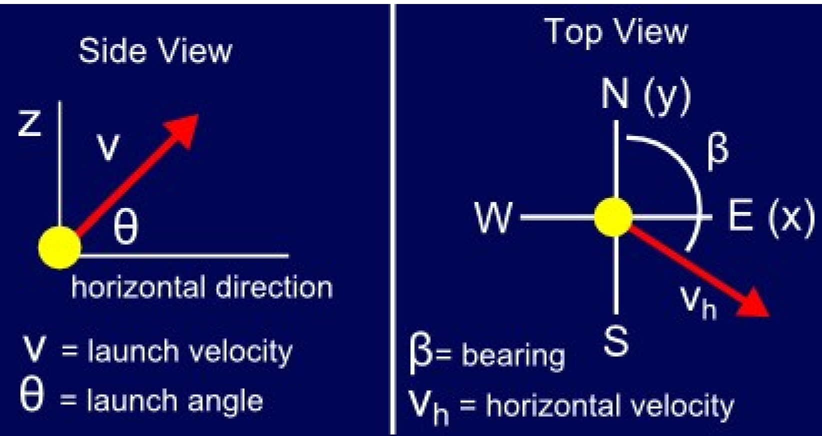
As you can see, the acceleration due to gravity and the horizontal component of the velocity are always constant, while the vertical component of velocity and the overall velocity change.

Units/Constants

All distances are measured in meters (m). Speeds are measured in meters per second (m/s). Bearing is measured in degrees east of north (clockwise). Launch Angle is measured in degrees counterclockwise from the ground. The acceleration of gravity is assumed to be 9.81 m/s^2.

Calculations

Projectile Battle takes place in 3 dimensions; The projectile has an x velocity where right is positive, a y velocity where up is positive and a z velocity where out of the screen is positive. The representation is actually pseudo 3d, with the z dimension being represented instead by the size of the projectile (it gets bigger as it goes higher up and closer to you). This means that for the calculations the projectile's vertical velocity is in the z direction and horizontal velocity is not necessarily in the x direction as you can assume in 2d, but it is in the x-y plane and so has components in both the x and y directions. This is illustrated in the picture below.



Bearing
Bearing is necessary for both easy and hard calculations. In order to calculate bearing, you must first calculate degrees counter clockwise from the x-axis. Use the inverse tangent ($\tan^{-1}$) or arc tangent (arctan), (the two mean the same thing) of $\Delta y/\Delta x$. Δx represents target x minus launch x, and Δy represents target y minus launch y.

$$\Delta x = x_{final} - x_{initial}$$
$$\Delta y = y_{final} - y_{initial}$$

$$\theta = \arctan\left(\frac{\Delta y}{\Delta x}\right)$$

The angle result that you get should be represented as degrees counter clockwise from the x-axis. The arctan will only give angles in the range of -90 to 90 degrees, so it's your job to figure out what the angle should actually be if it lies in the second or third quadrant. Another option you have is to use a calculator that has the atan2(Δy,Δx) function, which will do this quadrant determining work for you and give you the angle counter clockwise from the x-axis. On a TI-84 calculator you can use the function
R►Pθ(Δx,Δy). To get to R►Pθ, press [2ND][ANGLE][6]. Notice that atan2 has Δy before Δx and that R►Pθ has Δx before Δy.

After you have your angle in degrees, you need to convert it to bearing. Unlike the arctan where 0 degrees represents facing to the right along the x axis and positive angles rotate counterclockwise from 0, with bearing 0 degrees represents North and positive angles rotate clockwise from 0. You can use the formula below to convert from degrees counter clockwise from the x-axis to bearing.

$$bearing = 90 - angle$$

Bearing Tips: If it is easier for you, you may use negative bearings (counterclockwise from north). You can also enter any bearing greater than 360. Also, if your target lies along a line that's parallel to an axis, then just use the appropriate angle, there's no need to calculate it. For example, if the direction is due East, then the bearing is 90. In fact, if the original angle is 90 or 270 (bearing angle of 0 or 180, North or South), then the arctan formula won't even work due to Δx being 0.
Range
Range is necessary for easy calculations as an input to the game. For harder calculations it's a necessary intermediate calculation. In my opinion, it's also the easiest to calculate. Just use the distance formula:

$$Range = \sqrt{(\Delta x^2 + \Delta y^2)}$$

Alternatively, you can also use R►Pr(Δx,Δy) on TI-84 calculators. To get to it, press [2ND][ANGLE][5].

Launch Velocity
Launch velocity is only necessary for the harder calculations. Once you have range, you can use the formula below to calculate launch velocity, where v_0 = launch velocity, R = range, g = acceleration due to gravity (9.81 m/s^2), and θ_0 = launch angle. Ask your physics teacher to derive it for you.

$$v_0 = \sqrt{\frac{Rg}{\sin(2\theta_0)}}$$

As you can see, this equation requires launch angle to solve for launch velocity. If the angle is stuck, then use that angle, if it's not then you can pick any angle you want!

Tips: If both launch velocity and launch angle are unstuck, then it's usually easier to just use this equation (as opposed to the launch angle equation) with 45 degrees as the initial launch angle. The sin of 90 is 1, and so 45 degrees simplifies the equation a bit.

Launch Angle
Launch angle is only necessary for the harder calculations. This formula is the same as the formula provided for the above section, just solved for θ_0.

$$\theta_0 = \frac{1}{2}\arcsin\left(\frac{Rg}{v_0^2}\right)$$

Remember that arcsin is the same thing as $\sin^{-1}$. If the formula gives you an error, then you entered the formula incorrectly or the expression inside of the arcsin is outside the range of the arcsin function of -1 to 1. This means that with the provided velocity, there is no possible angle that can be used to have the projectile reach the target. (Think about it, if a projectile's launch velocity was 1 m/s could any angle make it reach a target 10 km away?) This is actually quite rare, and only happens if you want to aim at something really far away and your velocity is stuck at something relatively low. If this happens to you, sorry, you just have to aim for something closer.

Tips: I recommend only using this equation when launch velocity is stuck, because it can be slightly more restrictive (if Rg is outside of the range -1 to 1), and because it is just harder to enter into the calculator in my opinion.

Is this game realistic?

No. It neglects the effects air resistance, wind, and has perfectly rectangular boats with perfectly circular and consistent explosions. It's a turn based java game, so what did you expect? However, the distances and speeds are roughly proportional. i.e, if you have a fast computer then the projectile will accelerate downward at approximately the same rate as gravity, and it will move along the screen at speeds proportional to the grid scale.