

2D Game Physics - Part I - Angle & Velocity

- [benjamin805](#)

Table of Contents

[2D Game Physics - Part I - Angle & Velocity](#)

[Angle & Velocity](#)

[Drawing the object](#)

[Moving the object](#)

[Controlling the object](#)

Angle & Velocity

You can easily add basic physics to your games using these examples below. Original examples found at <http://www.roddev.com/tutorials/gamephysics/> .

In 2D games there are two position points that are used to place your sprite or drawing on the screen. They are known as the x axis and the y axis. In this example we will call them

object.x ' x axis

object.y ' y axis

We will use two variables called velocity.x and velocity.y to move the object on the screen.

velocity.x ' x axis movement increment

velocity.y ' y axis movement increment

We must be able to determine the direction and speed of the object moving on the screen. To do this we will use two variables called angle and speed.

angle ' direction of object in degrees (0-360)

speed ' The speed at witch the object moves

Using the Liberty BASIC commands [cos\(\)](#) and [sin\(\)](#) we will convert our angle value to give us another value that we will call the scale.x and scale.y.

```
scale.x ' cos() of angle  
scale.y ' sin() of angle
```

Since angle values can not be used directly with cos() and sin() we must first convert the angle to a [radian](#) value. We do this like so.

```
angle=angle/57.29577951
```

This gives us our radian value witch we can use with cos() and sin() to determine our scale values. The scale values will move the object in the direction we have chosen using our angle value.

```
scale.x=cos(angle)  
scale.y=sin(angle)
```

Once we have our scale values we can now combine them with our speed value to get our velocity values

```
velocity.x=speed*scale.x  
velocity.y=speed*scale.y
```

Remember that with each new direction the velocity values must be recalculated before moving the object.

Our code so far.

'2D Game Physics - Part I

```
object.x=400  
object.y=300  
  
angle=45  
r.angle=angle/57.29577951  
speed=5  
  
scale.x=cos(r.angle)  
scale.y=sin(r.angle)  
  
velocity.x=speed*scale.x  
velocity.y=speed*scale.y
```

Drawing the object

Now we will draw our object on the screen. For this example I will use a simple line to represent our object. By using the velocity values I can determine the line position and length. Below is an updated version of the code above.

'2D Game Physics - Part I

```
object.x=400
object.y=300

angle=45
r.angle=angle/57.29577951
speed=5

scale.x=cos(r.angle)
scale.y=sin(r.angle)

velocity.x=speed*scale.x
velocity.y=speed*scale.y

nomainwin

WindowWidth=800
WindowHeight=600
open "2D Game Physics - Part I" for graphics as #main
#main "trapclose [quit]"
#main "color red;size 5;down"
#main "line ";object.x;" ";object.y;" ";object.x+(velocity.x*2);" ";
object.y-(velocity.y*2)

wait

[quit]
close #main
end
```

Moving the object

We will now move our object on the screen in the direction we have chosen (45 degrees) until it reaches the edge of our window.

'2D Game Physics - Part I

```
object.x=400
```

```
object.y=300

angle=45
r.angle=angle/57.29577951
speed=5

scale.x=cos(r.angle)
scale.y=sin(r.angle)

velocity.x=speed*scale.x
velocity.y=speed*scale.y

nomainwin

WindowWidth=800
WindowHeight=600
open "2D Game Physics - Part I" for graphics_nsb as #main
#main "trapclose [quit]"
#main "color red;size 5;down"
#main "line ";object.x;" ";object.y;" ";object.x+(velocity.x*2);" ";
object.y-(velocity.y*2)

timer 100,[move]
wait

[move]
object.x=object.x+velocity.x    'increase object.x value
object.y=object.y-velocity.y    'increase object.y value

'check for edge of window
if object.y<10 then object.y=10
if object.y>WindowHeight-20 then object.y=WindowHeight-20
if object.x>WindowWidth-15 then object.x=WindowWidth-15
if object.x<10 then object.x=10

'draw object
#main "cls"
#main "line ";object.x;" ";object.y;" ";object.x+(velocity.x*2);" ";
object.y-(velocity.y*2)

wait
[quit]
close #main
end
```

Controlling the object

Let's now add some basic movement control to our object. In this example we will use the arrow keys on your keyboard to change the angle of the object which will change its direction. I have also created a sub called setangle which can be used to change the velocity values for each new direction we want to move in.

```
'2D Game Physics - Part I
```

```
object.x=400  
object.y=300
```

```
angle=45  
speed=10
```

```
call setangle velocity.x,velocity.y,angle,speed
```

```
nomainwin
```

```
WindowWidth=800
```

```
WindowHeight=600
```

```
open "2D Game Physics - Part I" for graphics_nsb as #main
```

```
#main "trapclose [quit]"
```

```
#main "when characterInput [getkey]"
```

```
#main "color red;size 5;down"
```

```
#main "line ";object.x;" ";object.y;" ";object.x+(velocity.x*2);" "  
object.y-(velocity.y*2)
```

```
timer 100,[move]
```

```
wait
```

```
[getkey]
```

```
key$ = Inkey$
```

```
    if len(key$) < 2 then
```

```
        else
```

```
            if right$(key$,1)=chr$(_VK_UP) then
```

```
                if moving=0 then
```

```
                    moving=1
```

```
                else
```

```
                    moving=0
```

```
                end if
```

```
            end if
```

```
            if right$(key$,1)=chr$(_VK_RIGHT) then
```

```
                angle=angle-5
```

```
        if angle<0 then angle=360
        call setangle velocity.x,velocity.y,angle,speed
    end if

    if right$(key$,1)=chr$(_VK_LEFT) then
        angle=angle+5
        if angle>360 then angle=0
        call setangle velocity.x,velocity.y,angle,speed
    end if

end if

wait

[move]
#main "setfocus"
scan

if moving=1 then
object.x=object.x+velocity.x    'increase object.x value
object.y=object.y-velocity.y    'increase object.y value

'check for edge of window
if object.y<10 then object.y=10
if object.y>WindowHeight-40 then object.y=WindowHeight-40
if object.x>WindowWidth-15 then object.x=WindowWidth-15
if object.x<10 then object.x=10
end if

'draw object
#main "cls"
#main "line ";object.x;" ";object.y;" ";object.x+(velocity.x*2);" ";
object.y-(velocity.y*2)

wait
[quit]
close #main
end

sub setangle byref velocity.x, byref velocity.y,angle,speed

r.angle=angle/57.29577951

scale.x=cos(r.angle)
scale.y=sin(r.angle)
```

```
velocity.x=speed*scale.x  
velocity.y=speed*scale.y
```

```
end sub
```

I hope you find this example helpful in your game programming and I hope to add more parts as I finish them.