

Lab 13 – Solution: Bouncing Ball

```
import turtle

turtle.shape("circle")
turtle.penup()

x, y = 0, 0
xdir, ydir = 3, 3
xlimit, ylimit = turtle.window_width() / 2, turtle.window_height() / 2

def move():
    global x, y, xdir, ydir

    x = x + xdir
    y = y + ydir

    if not -xlimit < x < xlimit:
        xdir = -xdir
    if not -ylimit < y < ylimit:
        ydir = -ydir

    turtle.goto(x, y)

    turtle.ontimer(move, 5)

turtle.ontimer(move, 10)

turtle.exitonclick()
```

Lab – 15 Solution: Free Hand Movement

```
import turtle
import random

x = 0
y = 0

xincr = 0
yincr = 0

dir = 'r'

def move():
    global x,y,xincr,yincr,food
    turtle.goto(x,y)
    x,y = turtle.xcor(), turtle.ycor()
    x+=xincr
    y+=yincr
    if (not -230<x<230) or (not -230<y<230):
        t = turtle.Turtle()
        t.color("red")
        t.write("Game Over!", font=("Arial",30,"normal"))
        turtle.done()
        return 0
        #xincr,yincr = 0,0

    turtle.ontimer(move,10)
```

```

def goeast():
    global dir,xincr,yincr
    if dir == 'u':
        turtle.rt(90)
    elif dir == 'l':
        turtle.rt(180)
    elif dir == 'd':
        turtle.lt(90)
    dir = 'r'
    xincr,yincr = 3,0
def gowest():
    global dir,xincr,yincr
    if dir == 'u':
        turtle.lt(90)
    elif dir == 'r':
        turtle.rt(180)
    elif dir == 'd':
        turtle.rt(90)
    dir = 'l'
    xincr,yincr = -3,0
def gonorth():
    global dir,xincr,yincr
    if dir == 'l':
        turtle.rt(90)
    elif dir == 'r':
        turtle.lt(90)
    elif dir == 'd':
        turtle.lt(180)
    dir = 'u'
    xincr,yincr = 0,3
def gosouth():
    global dir,xincr,yincr
    if dir == 'r':
        turtle.rt(90)
    elif dir == 'l':
        turtle.lt(90)
    elif dir == 'u':
        turtle.lt(180)
    dir = 'd'
    xincr,yincr = 0,-3

```

```

wn = turtle.Screen()
turtle.setup(500,500)
turtle.shape('turtle')
turtle.penup()
turtle.ontimer(move,10)

```

```

wn.onkey(goeast, 'Right')
wn.onkey(gowest, 'Left')
wn.onkey(gosouth, "Down")
wn.onkey(gonorth, "Up")
wn.listen()

```

```

turtle.done()

```

Space Invader Game

```

import turtle

```

```

import random

def goup():
    y=player.ycor()
    y +=10
    player.sety(y)
def godown():
    y = player.ycor()
    y -= 10
    player.sety(y)
def goleft():
    x = player.xcor()
    x -= 10
    player.setx(x)
def goright():
    x = player.xcor()
    x += 10
    player.setx(x)

def iscollide(t1,t2):
    xdiff = t1.xcor() - t2.xcor()
    ydiff = t1.ycor() - t2.ycor()
    if xdiff < 0:
        xdiff *= -1
    if ydiff < 0:
        ydiff *= -1
    if xdiff<40 and ydiff<40:
        return True
    return False
def firebullet():
    global bullet_state
    if bullet_state == 'ready':
        bullet_state = 'fire'
        x = player.xcor() # get the x location of the player
        y = player.ycor() + 10 # get the y location of the player
        bullet.setposition(x, y) # move the bullet to where the player is
        bullet.showturtle() # show the bullet

enemies = []
enemy_speed = 30

turtle.setup(500,500)
turtle.bgpic("bg.gif")
turtle.register_shape("invader.gif")
turtle.register_shape("ship.gif")

player = turtle.Turtle()
player.shape("ship.gif")
player.penup()
player.setpos(0,-150)

bullet = turtle.Turtle()
bullet.color('yellow')
bullet.penup()
bullet.speed(0)
bullet.setheading(90)
bullet.shapesize(2, 2)
bullet.hideturtle()
bullet_state = 'ready'

```

```

score = 0
sboard = turtle.Turtle()
sboard.penup()
sboard.speed(0)
sboard.setpos(-200,220)
sboard.color("yellow")
sboard.write("Score: "+str(score), font=("Arial",10,"bold"))
sboard.hideturtle()

wn = turtle.Screen()
wn.listen()
wn.onkey(goleft,"Left")
wn.onkey(goright,"Right")
wn.onkey(goup,"Up")
wn.onkey(godown,"Down")
wn.onkey(firebullet,"space")
for i in range(10):
    enemies.append(turtle.Turtle())
for enemy in enemies:
    x = random.randint(-200, 200)
    y = random.randint(0, 200)
    enemy.penup()
    enemy.setposition(x, y)
    enemy.shape("invader.gif")
while True:
    for enemy in enemies: # go through the enemies one by one
        if enemy.xcor() > 170: # check if the enemy hit the white line
            y = enemy.ycor() # get the current y location of the enemy
            y -= 40 # change the y location of the enemy by 40
            enemy.sety(y) # move it closer to us by y which is 40
            enemy.speed(0) # this makes the transition seems instant
            enemy.setx(-170) # move the enemy to the left side again

        if enemy.ycor() < -190: # check if the enemy hit the bottom white
line
            y = 200 # set y to the new location at the top of the screen
            enemy.speed(0) # this makes the transition seems instant
            enemy.sety(y) # move the enemy to the new location

    if iscollide(player,enemy):
        player.hideturtle()
        turtle.color('red')
        turtle.penup()
        turtle.setposition(0, 0)
        turtle.write("Game Over!", move=False, align="center",
font=("Arial", 35, "normal"))
        turtle.setposition(0, -50)
        turtle.done() # Stop Game
        break

    if iscollide(bullet, enemy) :
        bullet.hideturtle()
        bullet_state = 'ready'
        enemy.setposition(-200, 200)
        score += 1
        sboard.clear()
        sboard.penup()
        sboard.setposition(-220, 220)
        sboard.write("Your score is: "+str(score), font=("Arial", 10,
"normal"))

```

```

        sboard.hideturtle()

    if bullet_state == 'fire':
        y = bullet.ycor()
        y += 50
        bullet.sety(y)

    if bullet.ycor() > 200: # check if the bullet hit the top white
line
        bullet.hideturtle() # hide the billet
        bullet_state = 'ready' # change its state to ready to enable
the player to fire again

        x = enemy.xcor() # get the current x location of the enemy
        x += enemy_speed # change the x location of the enemy by its speed
which is 30
        enemy.setx(x) # move the enemy to its new x location

turtle.done()

```

Turtle Game:

```

import turtle
import random

x = 0
y = 0

xincr = 0
yincr = 0

def iscollision(t1,t2):
    xdiff = t1.xcor() - t2.xcor()
    ydiff = t1.ycor() - t2.ycor()
    if xdiff < 0:
        xdiff *= -1
    if ydiff < 0:
        ydiff *= -1

    if xdiff <= 20 and ydiff <= 20:
        return True

    return False

def move():
    global x,y,xincr,yincr,food
    turtle.goto(x,y)
    x,y = turtle.xcor(), turtle.ycor()
    x+=xincr
    y+=yincr
    if (not -230<x<230) or (not -230<y<230):
        t = turtle.Turtle()
        t.color("red")
        t.write("Game Over!", font=("Arial",30,"normal"))
        turtle.done()
        return 0

```

```

    #xincr,yincr = 0,0
    if iscollision(turtle,food):
        food.hideturtle()
        food.setpos(random.randint(-250, 250), random.randint(-250, 250))
        food.showturtle()
        w,h,ol = turtle.turtlesize()
        turtle.turtlesize(w+1,h+1,ol)

    turtle.ontimer(move,10)

def goeast():
    global xincr,yincr
    turtle.setheading(0)
    xincr,yincr = 3,0
def gowest():
    global xincr,yincr
    turtle.setheading(180)
    xincr,yincr = -3,0
def gonorth():
    global xincr,yincr
    turtle.setheading(90)
    xincr,yincr = 0,3
def gosouth():
    global xincr,yincr
    turtle.setheading(270)
    xincr,yincr = 0,-3

food = turtle.Turtle()
food.color("green")
food.shape("circle")
food.penup()
food.speed(0)
food.setpos(random.randint(-250,250), random.randint(-250,250))

wn = turtle.Screen()
turtle.setup(500,500)
turtle.shape('turtle')
turtle.penup()
turtle.ontimer(move,10)
turtle.shapesize(1.0,1.0,1)
wn.onkey(goeast, 'Right')
wn.onkey(gowest, 'Left')
wn.onkey(gosouth, "Down")
wn.onkey(gonorth, "Up")
wn.listen()

turtle.done()

```