

```

from turtle import *
import random
import math

screen = Screen()

screenMinX = -screen.window_width()/2
screenMinY = -screen.window_height()/2
screenMaxX = screen.window_width()/2
screenMaxY = screen.window_height()/2

screen.setworldcoordinates(screenMinX,screenMinY,screenMaxX,screenMaxY)
screen.bgcolor("black")

penup()
ht()
speed(0)
goto(0, screenMaxY - 20)
color('red')
write("Asteroids!!", align="center", font=("Arial",20))
goto(0, screenMaxY - 33)
write("Use the arrow keys to move and 'space bar' to fire", align="center")
goto(0, 0)
color("lightblue")

class Bullet(Turtle):
    def __init__(self,screen,x,y,heading):
        Turtle.__init__(self)
        self.speed(0)
        self.penup()
        self.goto(x,y)
        self.seth(heading)
        self.screen = screen
        self.color('yellow')
        self.max_distance = 500
        self.distance = 0
        self.delta = 20
        self.shape("bullet")

    def move(self):
        self.distance = self.distance + self.delta
        self.forward(self.delta)
        if self.done():
            self.reset()

    def getRadius(self):
        return 4

    def blowUp(self):
        self.goto(-300,0)

    def done(self):
        return self.distance >= self.max_distance

class Asteroid(Turtle):
    def __init__(self,screen,dx,dy,x,y,size):
        Turtle.__init__(self)
        self.speed(0)

```

```

self.penup()
self.goto(x,y)
self.color('lightgrey')
self.size = size
self.screen = screen
self.dx = dx
self.dy = dy
self.shape("rock" + str(size))

def getSize(self):
    return self.size

def getDX(self):
    return self.dx

def getDY(self):
    return self.dy

def setDX(self,dx):
    self.dx = dx

def setDY(self,dy):
    self.dy = dy

def move(self):
    x = self.xcor()
    y = self.ycor()

    x = (self.dx + x - screenMinX) % (screenMaxX - screenMinX) + screenMinX
    y = (self.dy + y - screenMinY) % (screenMaxY - screenMinY) + screenMinY

    self.goto(x,y)

def blowUp(self):
    self.goto(-300,0)

def getRadius(self):
    return self.size * 10 - 5

class SpaceShip(Turtle):
    def __init__(self,screen,dx,dy,x,y):
        Turtle.__init__(self)
        self.speed(0)
        self.penup()
        self.color("white")
        self.goto(x,y)
        self.dx = dx
        self.dy = dy
        self.screen = screen
        self.bullets = []
        screen.addshape("rocketship.gif")
        self.shape("rocketship.gif")

    def move(self):
        x = self.xcor()
        y = self.ycor()

        x = (self.dx + x - screenMinX) % (screenMaxX - screenMinX) + screenMinX
        y = (self.dy + y - screenMinY) % (screenMaxY - screenMinY) + screenMinY

```

```

self.goto(x,y)

def powPow(self, asteroids):
    dasBullets = []
    for bullet in self.bullets:
        bullet.move()
        hit = False
        for asteroid in asteroids:
            if intersect(asteroid, bullet):
                asteroids.remove(asteroid)
                asteroid.blowUp()
                bullet.blowUp()
                hit = True
        if (not bullet.done() and not hit):
            dasBullets.append(bullet)

self.bullets = dasBullets

def fireBullet(self):
    self.bullets.append(Bullet(self.screen, self.xcor(), self.ycor(), self.heading()))

def fireEngine(self):
    angle = self.heading()
    x = math.cos(math.radians(angle))
    y = math.sin(math.radians(angle))
    self.dx = self.dx + x
    self.dy = self.dy + y

def turnTowards(self,x,y):
    if x < self.xcor():
        self.left(7)
    if x > self.xcor():
        self.right(7)

def getRadius(self):
    return 10

def getDX(self):
    return self.dx

def getDY(self):
    return self.dy

def intersect(object1,object2):
    dist = math.sqrt((object1.xcor() - object2.xcor())**2 + (object1.ycor() - object2.ycor())**
2)

    radius1 = object1.getRadius()
    radius2 = object2.getRadius()

    if dist <= radius1+radius2:
        return True
    else:
        return False

screen.register_shape("rock3",((-20, -16),(-21, 0), (-20,18), (0,27), (17,15), (25,0), (16,-15), (
0,-21)))

```

```

screen.register_shape("rock2",((-15, -10),(-16, 0), (-13,12),(0,19),(12,10),(20,0),(12,-10),(
0,-13)))
screen.register_shape("rock1",((-10,-5),(-12,0),(-8,8),(0,13),(8,6),(14,0),(12,0),(8,-6),(0,-
7)))
screen.register_shape("ship",((-10,-10),(0,-5),(10,-10),(0,10)))
screen.register_shape("bullet",((-2,-4),(-2,4),(2,4),(2,-4)))

```

```

ship = SpaceShip(screen,0,0,(screenMaxX-screenMinX)/2+screenMinX,(screenMaxY-screenMinY)/2 +
screenMinY)

```

```

asteroids = []

```

```

for k in range(5):
    dx = random.random() * 6 - 3
    dy = random.random() * 6 - 3
    x = random.random() * (screenMaxX - screenMinX) + screenMinX
    y = random.random() * (screenMaxY - screenMinY) + screenMinY

```

```

    asteroid = Asteroid(screen,dx,dy,x,y,random.randint(1,3))

```

```

    asteroids.append(asteroid)

```

```

def play():

```

```

    ship.move()

```

```

    gameover = False

```

```

    for asteroid in asteroids:
        asteroid.move()
        if intersect(ship,asteroid):
            write("BOOM!!!",font=("Arial",30),align="center")
            gameover = True

```

```

    ship.powPow(asteroids)

```

```

    screen.update()

```

```

    if not asteroids:
        color('green')
        write("You Win!",font=("Arial",30),align="center")

```

```

    if not gameover:
        screen.ontimer(play, 30)

```

```

bullets = []

```

```

def turnLeft():
    ship.left(7)

```

```

def turnRight():
    ship.right(7)

```

```

def go():
    ship.fireEngine()

```

```

def fire():
    ship.fireBullet()

```

```

ht()

```

```
screen.tracer(0);
```

```
screen.onkey(turnLeft, 'left')  
screen.onkey(turnRight, 'right')  
screen.onkey(go, 'up')  
screen.onkey(fire, 'space')  
screen.listen()
```

```
play()
```