```
import turtle
import time
import random


wn=turtle.Screen()
game=True
reset=True
gameScore=-1
pixLis=[]
snakeLis=[]
turtLis=[]
counter=0
crtDir="right"
valLis=[]
pstDir="right"
checker=[0,0,0,0]

def lose():
  global gameScore
  global wn
  print "You lose!"
  print "You ate..."
  print gameScore
  print "...fruits!"
  print "Press up to play again!"
  game=False
  display()
  wn.onkey(start, "Up")
  while game==False:
    wn.listen()


def fruit():
  global pixLis
  global reset
  global gameScore
  gameScore+=1
  while reset==True:
    g=random.randint(0,255)
    if pixLis[g]==0:
      pixLis[g]=-1
      reset=False
  reset=True
```

```
def snakeLen():
  global crtDir
  global pixLis
  global snakeLis
  for x in pixLis:
    if x==len(snakeLis):
      if crtDir=="right":
        pixLis[pixLis.index(x)+16]=len(snakeLis)+1
        snakeLis.append(pixLis.index(len(snakeLis)+1))
        display()
      elif crtDir=="left":
        pixLis[pixLis.index(x)-16]=len(snakeLis)+1
        snakeLis.append(pixLis.index(len(snakeLis)+1))
        display()
      elif crtDir=="up":
        pixLis[pixLis.index(x)+1]=len(snakeLis)+1
        snakeLis.append(pixLis.index(len(snakeLis)+1))
        display()
      elif crtDir=="down":
        pixLis[pixLis.index(x)-1]=len(snakeLis)+1
        snakeLis.append(pixLis.index(len(snakeLis)+1))
        display()
```

```
def Fchecker():
  count=0
  for x in pixLis:
    if x==-1:
      count+=1
  if count==1:
    pass
  else:
    snakeLen()
    fruit()
```

```
def valLispop():
  global valLis
  q=-1
  b=0
  for x in range (256):
    permX="X"
    permY="Y"
    q+=1
    if q==16:
      q=0
      b+=1
    b=str(b)
    q=str(q)
    y=permX+b+permY+q
    b=int(b)
    q=int(q)
    valLis.append(y)
```

```
def pisLispop():
  global pixLis
  for x in range (256):
    pixLis.append(0)
```

```
def pixListrans():
  global snakeLis
  global pixLis
  snakeLis.append(218)
  snakeLis.append(234)
  snakeLis.append(250)
  pixLis[snakeLis[0]]=snakeLis.index(snakeLis[0])+1
  pixLis[snakeLis[1]]=snakeLis.index(snakeLis[1])+1
  pixLis[snakeLis[2]]=snakeLis.index(snakeLis[2])+1
```

```
def firPos():
  checker[0]=1
def secPos():
  checker[1]=1
def thirPos():
  checker[2]=1
def fourPos():
  checker[3]=1


def display():
  global pixLis
  global turtLis
  for x in range(0,256):
    if pixLis[x]>0:
      z=turtLis[x]
      z.color("red")
    elif pixLis[x]==0:
      z=turtLis[x]
      z.color("black")
    elif pixLis[x]==-1:
      z=turtLis[x]
      z.color("green")




def down(num,gate=True):
  global pixLis
  global checker
  global valLis
  pos=pixLis.index(num)
  x=valLis[pos]
  y=x[x.index("Y")+1:]
  k=x[x.index("X")+1:x.index("Y")]
  y=int(y)
  k=int(k)
  y+=1
  if y==16:
    lose()
    y=0
    y=str(y)
    k+=1
    k=str(k)
    g="X"+k+"Y"+y
  else:
    y=str(y)
    c=x[0:x.index("Y")+1]
    g=c+y
  if k=="16":
    lose()
    k="0"
    g="X"+k+"Y"+y
  h=valLis.index(g)
  pixLis[pos]=0
  pixLis[h]=num
  for x in pixLis:
    temp=num+1
    if x==temp and gate==True:
      down(x)
  pixLis[h]=num
  Fchecker()
  display()

def up(num,gate=True):
  global pixLis
  global checker
  global valLis
  pos=pixLis.index(num)
  x=valLis[pos]
  y=x[x.index("Y")+1:]
  k=x[x.index("X")+1:x.index("Y")]
  y=int(y)
  k=int(k)
  y-=1
  if y==-1:
    lose()
    y=15
    y=str(y)
    k-=1
    k=str(k)
    g="X"+k+"Y"+y
  elif y!=-1:
    y=str(y)
    c=x[0:x.index("Y")+1]
    g=c+y
  if k=="-1":
    lose()
    k="15"
    g="X"+k+"Y"+y
  h=valLis.index(g)
  pixLis[pos]=0
  pixLis[h]=num
  for x in pixLis:
    if x==num+1 and gate==True:
      up(x)
  pixLis[h]=num
  Fchecker()
  display()

def right(num,gate=True):
  global pixLis
  global checker
  global valLis
  pos=pixLis.index(num)
  x=valLis[pos]
  y=x[x.index("Y")+1:]
  k=x[x.index("X")+1:x.index("Y")]
  y=int(y)
  k=int(k)
  k-=1
  if k==-1:
    lose()
    k=15
    k=str(k)
    y=str(y)
    g="X"+k+"Y"+y
  else:
    k=str(k)
    y=str(y)
    g="X"+k+"Y"+y
  h=valLis.index(g)
  pixLis[pos]=0
  pixLis[h]=num
  for x in pixLis:
    if x==num+1 and gate==True:
      right(x)
  pixLis[h]=num
  Fchecker()
  display()

def left(num,gate=True):
  global pixLis
  global checker
  global valLis
  pos=pixLis.index(num)
  x=valLis[pos]
  y=x[x.index("Y")+1:]
  k=x[x.index("X")+1:x.index("Y")]
  y=int(y)
  k=int(k)
  k+=1
  if k==16:
    lose()
    k=0
    k=str(k)
    y=str(y)
    g="X"+k+"Y"+y
  else:
    k=str(k)
    y=str(y)
    g="X"+k+"Y"+y
  h=valLis.index(g)
  pixLis[pos]=0
  pixLis[h]=num
  for x in pixLis:
    if x==num+1 and gate==True:
      left(x)
  pixLis[h]=num
  Fchecker()
  display()



def twodown():
  global pstDir
  global checker
  global runtime
  global crtDir
  global snakeLis
  if pstDir=="up":
    count=1
    for y in range(len(snakeLis)-1):
      for x in range(count):
        down(x+1,False)
      count+=1
      up(count)
  elif pstDir=="left":
    count=1
    for y in range(len(snakeLis)-1):
      for x in range(count):
        down(x+1,False)
      count+=1
      left(count)
  elif pstDir=="right":
    count=1
    for y in range(len(snakeLis)-1):
      for x in range(count):
        down(x+1,False)
      count+=1
      right(count)
  while game==True:
    down(1)
    wn.onkey(firPos, "Up")
    wn.onkey(secPos, "Left")
    wn.onkey(thirPos, "Right")
    wn.onkey(fourPos, "Down")
    wn.listen()
    if checker[1]==1 or checker[2]==1:
      pstDir="down"
      break
    else:
      pass
  if checker[1]==1:
    checker=[0,0,0,0]
    crtDir="left"
    twoleft()
  elif checker[2]==1:
    checker=[0,0,0,0]
    crtDir="right"
    tworight()
  else:
    pass


def twoup():
```

```
global pstDir
global checker
global runtime
global crtDir
global snakeLis
if pstDir=="down":
  for x in pixLis:
    count=1
  for y in range(len(snakeLis)-1):
    for x in range(count):
      up(x+1,False)
    count+=1
    down(count)
elif pstDir=="left":
  count=1
  for y in range(len(snakeLis)-1):
    for x in range(count):
      up(x+1,False)
    count+=1
    left(count)
elif pstDir=="right":
  count=1
  for y in range(len(snakeLis)-1):
    for x in range(count):
      up(x+1,False)
    count+=1
    right(count)
while game==True:
  up(1)
  wn.onkey(firPos, "Up")
  wn.onkey(secPos, "Left")
  wn.onkey(thirPos, "Right")
  wn.onkey(fourPos, "Down")
  wn.listen()
  if checker[1]==1 or checker[2]==1:
    pstDir="up"
    break
if checker[1]==1:
  checker=[0,0,0,0]
  crtDir="left"
  twoleft()
elif checker[2]==1:
  checker=[0,0,0,0]
  crtDir="right"
```

```
    tworight()
  else:
    pass


def twoleft():
  global pstDir
  global checker
  global crtDir
  if pstDir=="down":
    count=1
    for y in range(len(snakeLis)-1):
      for x in range(count):
        left(x+1,False)
      count+=1
      down(count)
  elif pstDir=="up":
    count=1
    for y in range(len(snakeLis)-1):
      for x in range(count):
        left(x+1,False)
      count+=1
      up(count)
  elif pstDir=="right":
    count=1
    for y in range(len(snakeLis)-1):
      for x in range(count):
        left(x+1,False)
      count+=1
      right(count)
  while game==True:
    left(1)
    wn.onkey(thirPos,"Left")
    wn.onkey(firPos, "Down")
    wn.onkey(secPos, "Up")
    wn.onkey(fourPos, "Right")
    wn.listen()
    if checker[0]==1 or checker[1]==1:
      pstDir="left"
      break
    else:
      pass
  if checker[0]==1:
    checker=[0,0,0,0]
    crtDir="down"
    twodown()
  elif checker[1]==1:
    checker=[0,0,0,0]
    crtDir="up"
    twoup()
  else:
    pass

def tworight():
  global pstDir
  global checker
  global counter
  global runtime
  global crtDir
  if pstDir=="down":
    count=1
    for y in range(len(snakeLis)-1):
      for x in range(count):
        right(x+1,False)
      count+=1
      down(count)
  elif pstDir=="up":
    count=1
    for y in range(len(snakeLis)-1):
      for x in range(count):
        right(x+1,False)
      count+=1
      up(count)
  elif pstDir=="left":
    count=1
    for y in range(len(snakeLis)-1):
      for x in range(count):
        right(x+1,False)
      count+=1
      left(count)
  while game==True:
    counter+=1
    right(1)
    if counter==3:
      wn.update()
      wn.tracer(50,1)
    wn.onkey(thirPos,"Left")
    wn.onkey(firPos, "Down")
    wn.onkey(secPos, "Up")
    wn.onkey(fourPos,"Right")
    wn.listen()
    if checker[0]==1 or checker[1]==1:
      pstDir="right"
      break
  if checker[0]==1:
    checker=[0,0,0,0]
    crtDir="down"
    twodown()
  elif checker[1]==1:
    checker=[0,0,0,0]
    crtDir="up"
    twoup()


def turtLispop():
  global turtLis
  for x in range(256):
    turtLis.append(turtle.Turtle())




def printer():
  lsIn=0
  mgVal=0
  inc=0
  dbInc=1
  snInc=0
  for x in range(0,16):
    snInc=0
    for x in turtLis[inc*16:dbInc*16]:
      x.penup()
      x.setpos((180-mgVal),180-(snInc*20))
      snInc+=1
    inc+=1
    dbInc+=1
    lsIn+=1
    mgVal=lsIn*20
```

```
def start():
  print "Use arrow keys for direction!"
  print "!Please wait for the snake to straighten up before making your next move..."
  print "...otherwise the snake may not move in the intended direction!"
  global wn
  global game
  global reset
  global gameScore
  global pixLis
  global snakeLis
  global turtLis
  global counter
  global crtDir
  global valLis
  global pstDir
  global checker
  wn=turtle.Screen()
  game=True
  reset=True
  gameScore=-1
  pixLis=[]
  snakeLis=[]
  turtLis=[]
  counter=0
  crtDir="right"
  valLis=[]
  pstDir="right"
  checker=[0,0,0,0]
  valLispop()
  pisLispop()
  pixListrans()
  turtLispop()
  wn.tracer(0,0)
  printer()
  fruit()
  display()
  tworight()
  wn.listen()

start()
```