

```
#
# Machine Problem 5
# CS 1300
# Bruce Miller
#
# Description: This script will use the cs1graphics module to create a graphics
# scene and then animate the scene. The initial scene has a farm-themed background
# and a chicken.
# The chicken pecks the ground until it gets a worm, as a pig comes on screen. The
# pig begins to fly,
# and the chicken watches it fly higher and higher.
#
```

```
from cs1graphics import *
from time import sleep
```

```
def makeHills():
    #
    # No parameters, returns 2 big green circles in background
    #

    hills = Layer()

    hill1 = Circle(1250, Point(1200, 1550))
    hill1.setFillColor((0, 50, 0))
    hill1.setBorderColor((0, 50, 0))
    hills.add(hill1)

    hill2 = Circle(1000, Point(400, 1350))
    hill2.setFillColor((0, 75, 0))
    hill2.setBorderColor((0, 75, 0))
    hills.add(hill2)

    return hills
```

```
def makeGrass():
    #
    # No parameters, returns green rectangle at bottom screen
    #

    grass = Rectangle(WIDTH, HEIGHT // 2, Point(WIDTH // 2, HEIGHT * 3 // 4))
    grass.setFillColor('darkGreen')
    grass.setBorderColor('darkGreen')
    return grass
```

```
def makeSun():
    #
    # No parameters, returns yellow circle at top-left of screen
    #

    sun = Circle(50, Point(100, 100))
    sun.setFillColor('yellow')
```

```
sun.setBorderColor('yellow')
return sun
```

```
def makeFence():
    #
    # Define amount of fencePosts, fence boardWidth, etc.
    # Add fenceposts to layer from calculated position.
    # No parameters, returns fence layer.
    #
    fencePosts = 9
    boardWidth = HEIGHT // 30
    boardHeight = HEIGHT * 3 // 20
    fenceGapWidth = (WIDTH - (fencePosts * boardWidth)) // (fencePosts + 1)

    fence = Layer()
    mainBoard = Rectangle(WIDTH, boardWidth, Point(WIDTH // 2, HEIGHT * 4 // 10))
    mainBoard.setFillColor('brown')
    fence.add(mainBoard)
    for post in range(fencePosts):
        boardPos = Point(fenceGapWidth * (post + 1) + boardWidth * (post + 0.5),
HEIGHT // 2 - boardHeight // 2)
        postObject = Rectangle(boardWidth, boardHeight, boardPos)
        postObject.setFillColor('brown')
        fence.add(postObject)

    return fence
```

```
def makeBarn():
    #
    # Construct barn from shapes and lines
    # No parameters, returns barn layer
    #
    barn = Layer()

    barnBack = Square(250, Point(1200, 350))
    barnBack.setFillColor('red')
    barn.add(barnBack)

    barnRoof = Polygon(Point(1075, 225), Point(1325, 225), Point(1200, 100))
    barnRoof.setFillColor('red')
    barn.add(barnRoof)

    gutter = Path(Point(1050, 250), Point(1200, 100), Point(1350, 250))
    gutter.setBorderColor('white')
    gutter.setBorderWidth(20)
    barn.add(gutter)

    roofFlat = Path(Point(1075, 225), Point(1325, 225))
    roofFlat.setBorderColor('white')
    roofFlat.setBorderWidth(20)
```

```

    barn.add(roofFlat)

    door = Path(Point(1125, 465), Point(1125, 290), Point(1275, 290), Point(1275,
465),
                Point(1125, 465), Point(1275, 290), Point(1125, 290), Point(1275,
465))
    door.setBorderColor('white')
    door.setBorderWidth(20)
    barn.add(door)

    return barn

def makeClouds():
    #
    # Construct cloud layer edge via circles and fill with ellipse
    # No parameters, returns cloud layer
    #
    cloud = Layer()
    myWorld.add(cloud)

    puff = Ellipse(80, 80, Point(50, 5))
    puff.setFillColor('white')
    puff.setBorderColor('lightGray')
    cloud.add(puff)

    puffPositions = [[0,0], [15,15], [30,25], [50,30], [70,25], [85,15], [100, 0],
[100,-20],
                    [0,-20], [15,-35], [30,-45], [50,-50], [70,-45], [85,-35],
[100,-20]]
    for centerPoint in puffPositions:
        puff = Circle(20, Point(centerPoint[0], centerPoint[1]))
        puff.setFillColor('white')
        puff.setBorderColor('lightGray')
        cloud.add(puff)

    cloud.move(0, 50)

    for i in range(15):
        cloud = cloud.clone()
        cloud.move(100, 0)
        myWorld.add(cloud)

def makeMud():
    #
    # Construct mud layer with ellipses
    # No parameters, returns mud layer
    #
    mud = Layer()

```

```
spot = Ellipse(200, 100, Point(400, 700))
spot.setFillColor((54, 40, 26))
spot.setBorderColor((54, 40, 26))
mud.add(spot)
```

```
spot1 = Ellipse(125, 60, Point(450, 750))
spot1.setFillColor((54, 40, 26))
spot1.setBorderColor((54, 49, 26))
mud.add(spot1)
```

```
spot2 = Ellipse(200, 100, Point(395, 705))
spot2.setFillColor((120, 62, 34))
spot2.setBorderColor((54, 40, 26))
mud.add(spot2)
```

```
spot3 = Ellipse(125, 60, Point(445, 755))
spot3.setFillColor((120, 62, 34))
spot3.setBorderColor((54, 49, 26))
mud.add(spot3)
```

```
return mud
```

```
def makeChicken():
    #
    # Construct chicken head and body and connect together
    # No parameters, returns: whole chicken layer, chicken head
    #
    chicken = Layer()

    chickBody = Circle(25)
    chickBody.setFillColor('white')
    chicken.add(chickBody)

    wing = Polygon(Point(10, -5), Point(-10, 0), Point(10, 5))
    chicken.add(wing)

    leg1 = Path(Point(-10, 20), Point(-10, 30), Point(-15, 35), Point(-10, 30),
                Point(-10, 37), Point(-10, 30), Point(-4, 35))
    leg1.setBorderColor((191, 105, 0))
    leg1.setBorderWidth(3)
    chicken.add(leg1)
    leg2 = leg1.clone()
    leg2.move(20, 0)
    chicken.add(leg2)

    headRig = Layer()

    chickHead = Circle(15, Point(25, -25))
```

```

chickHead.setFillColor('white')
headRig.add(chickHead)

beak = Polygon(Point(35, -30), Point(55, -25), Point(35, -20))
beak.setFillColor('yellow')
headRig.add(beak)
headRig.rotate(40)

eye = Circle(0.55, Point(30, -25))
headRig.add(eye)

chicken.add(headRig)

chicken.move(1000, 500)

return chicken, headRig

```

```

def makePig():
    #
    # Construct pig layer, setup legs for rotation
    # No parameters, returns: pig layer, two legs, & tail
    #
    pig = Layer()

    body = Ellipse(100, 50)
    body.setFillColor('pink')
    body.setBorderColor('pink')
    pig.add(body)

    tail = Path(Point(-40, 0), Point(-70, -30))#, Point(-80, -20), Point(-, -10))
    tail.setBorderColor((255, 132, 200))
    tail.setBorderWidth(3)
    pig.add(tail)

    head = Ellipse(50, 40, Point(40, -20))
    head.setFillColor('pink')
    pig.add(head)

    nose = Ellipse(10, 8, Point(40, -20))
    pig.add(nose)
    nostril = Circle(0.55, Point(38, -20))
    pig.add(nostril)
    nostril = nostril.clone()
    nostril.move(4, 0)
    pig.add(nostril)

    eye = Circle(0.55, Point(30, -30))
    pig.add(eye)
    eye = eye.clone()
    eye.move(20, 0)

```

```

pig.add(eye)

mouth = Circle(0.55, Point(40, -10))
pig.add(mouth)

leg1 = Rectangle(10, 20, Point(-40, 20))
leg1.setFillColor('pink')
leg1.adjustReference(0, -10)
pig.add(leg1)

leg2 = leg1.clone()
leg2.move(80, 0)
pig.add(leg2)

return pig, leg1, leg2, tail

def swingObject(object, direction, speed, rotatePos, limit):
    #
    # rotate / "swing" object back and forth according to parameters
    # Takes parameters: object to rotate, which direction it is rotating,
    # speed of rotation, position of rotation, and limit of rotation.
    # Returns: rotation direction (changes if object rotates past its limit),
    position in rotation
    #
    if direction == 'right':
        object.rotate(speed)
        rotatePos += speed
    elif direction == 'left':
        object.rotate(-speed)
        rotatePos -= speed
    if rotatePos >= limit:
        direction = 'left'
    elif rotatePos <= -limit:
        direction = 'right'

    return direction, rotatePos

### Setup Canvas
WIDTH = 1600
HEIGHT = 900
myWorld = Canvas(WIDTH, HEIGHT, 'skyBlue', 'My World')
myWorld.add(makeSun())
makeClouds()
myWorld.add(makeHills())
myWorld.add(makeGrass())
myWorld.add(makeFence())
myWorld.add(makeBarn())
myWorld.add(makeMud())

```

```

chicken, headRig = makeChicken()
myWorld.add(chicken)

pig, leg1, leg2, tail = makePig()
myWorld.add(pig)
pig.move(-100, 700)

### Animate
timeDelay = 1 / 30 # 30 fps
sceneLength = 0 # seconds

chickHeadDir = 'right'
chickHeadRotatePos = 0

chickWaddleDir = 'right'
chickWaddlePos = 0

leg1Dir = 'right'
leg1RotatePos = 0

leg2Dir = 'left'
leg2RotatePos = 0

tailDir = 'right'
tailRotatePos = 0

needsWorm = True

loopFrames = True
while loopFrames:
    if not sceneLength <= 2 and sceneLength <= 3:
        pig.move(10, 0)
        pig.move(0, -10)
        tailDir, tailRotatePos = swingObject(tail, tailDir, 5, tailRotatePos, 10)
        chickWaddleDir, chickWaddlePos = swingObject(chicken, chickWaddleDir, 1,
chickWaddlePos, 2)
        chickHeadDir, chickHeadRotatePos = swingObject(headRig, chickHeadDir, 2,
chickHeadRotatePos, 15)
        leg1.rotate(-60)
        leg2.rotate(60)
    elif not sceneLength <= 3 and sceneLength <= 4:
        pig.move(10, 0)
        pig.move(0, 5)
        tailDir, tailRotatePos = swingObject(tail, tailDir, 5, tailRotatePos, 10)
        chickWaddleDir, chickWaddlePos = swingObject(chicken, chickWaddleDir, 1,
chickWaddlePos, 2)
        chickHeadDir, chickHeadRotatePos = swingObject(headRig, chickHeadDir, 2,
chickHeadRotatePos, 15)
        leg1.rotate(-60)
        leg2.rotate(60)

```

```

elif not sceneLength <= 4 and sceneLength <= 4.25:
    pig.move(0, -40)
    tailDir, tailRotatePos = swingObject(tail, tailDir, 5, tailRotatePos, 10)
    headRig.rotate(-11)
    leg1.rotate(-60)
    leg2.rotate(60)
elif not sceneLength <= 4.25 and sceneLength <= 4.75:
    pig.move(0, -40)
    tailDir, tailRotatePos = swingObject(tail, tailDir, 5, tailRotatePos, 10)
    headRig.rotate(-0.25)
    leg1.rotate(-60)
    leg2.rotate(60)
elif not sceneLength <= 4.75:
    headRig.rotate(-0.25)
else:
    pig.move(10, 0)
    chickWaddleDir, chickWaddlePos = swingObject(chicken, chickWaddleDir, 1,
chickWaddlePos, 2)
    chickHeadDir, chickHeadRotatePos = swingObject(headRig, chickHeadDir, 2,
chickHeadRotatePos, 15)
    leg1Dir, leg1RotatePos = swingObject(leg1, leg1Dir, 15, leg1RotatePos, 45)
    leg2Dir, leg2RotatePos = swingObject(leg2, leg2Dir, 15, leg2RotatePos, 45)

    if sceneLength >= 1.75 and needsWorm:
        worm = Path(Point(0, 0), Point(0, 5), Point(5, -5), Point(10, -5), Point(10,
-10))
        worm.setBorderColor('pink')
        worm.setBorderWidth(2)
        worm.move(55, -30)
        headRig.add(worm)
        needsWorm = False

    sleep(timeDelay)
    sceneLength += timeDelay

    if sceneLength >= 9.5:
        loopFrames = False

```