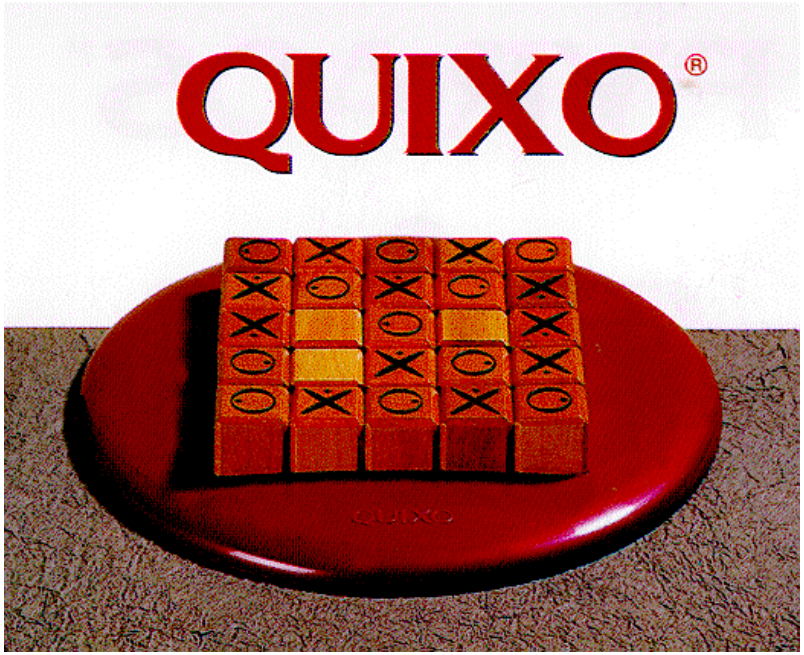
OPERATIONS ON COLLECTIONS & PROJECT PREVIEW

CSSE 120—Rose Hulman Institute of Technology

Project preview

- You will be implementing Quixo
  - http://www.educationallearninggames.com/how-to-play-quixo-game-rules.asp
- Team size: 3 students
  - You should take/have taken "Team Preference Survey" on Angel
- We will actually begin the project next session
- Due: presentations will be done end of week 7
- Today's homework will help you prepare for project

Game that you will implement



Lists are sequences

- All of Python's built-in sequence operations apply

Operator	Meaning
<seq> + <seq>	Concatenation
<seq> * <int-expr>	Repetition
<seq>[ ]	Indexing
len(<seq>)	Length
<seq>[:]	Slicing
for <var> in <seq>:	Iteration
<expr> in <seq>	Membership check (Returns a Boolean)

List-specific methods

- These methods can also be applied to list objects
- Some of them mutate a list

Method	Meaning
<list>.append(x)	Add element x to end of list
<list>.sort()	Sort (order) the list. A comparison function may be passed as a parameter
<list>.reverse()	Reverse the list
<list>.index(x)	Return index of first occurrence of x
<list>.insert(i, x)	Insert x into list at index i
<list>.count(x)	Return number of occurrences of x in list
<list>.remove(x)	Delete first occurrence of x in list
<list>.pop(i)	Delete i^{th} element of list and return its value

What can we do with lists?

- Do the same thing to each object in a list
- Find the largest number in a list of numbers.
- Find the second largest element.
- Find the point in a list that is farthest away from a given point.
- Find the point in a list which, when chosen as the center, can enclose all of the points in the smallest circle

Experimenting with list objects

```
colorList = [color_rgb(r,0,255-r) for r in range (0,255,2)] + \
            [color_rgb(255-r,r,0) for r in range (0,255,2)] + \
            [color_rgb(r,255-r,r) for r in range (0,255,2)] + \
            [color_rgb(255,r,255-r) for r in range (0,255,2)]

def moveAllElementsBy(list, dx, dy):
    for obj in list:
        obj.move(dx, dy)

def colorAll(list, color):
    for obj in list:
        obj.setFill(color)

def moveThoseColors():
    win = GraphWin("", 950, 600)
    rectList = []
    for i in range(5):
        rect = Rectangle(Point(i*50, 10), Point(i*50+40, 50))
        rect.draw(win)
        rectList.append(rect)

    for c in colorList:
        time.sleep(.02)
        moveAllElementsBy(rectList, 1, 1)
        colorAll(rectList, c)

    time.sleep(1)
```

Watch the demo

The first two functions are examples of doing the same thing to each element of a list.

Write and test these functions

1.
```
def doubleAll(list):
    """ returns a list of numbers that are twice
        those in the original list. """
```

2.
```
def largestInList(numList):  # A nonempty list of numbers
    """ returns the largest number in the list. """
```

3.
```
def secondLargest(numList):
    # numList contains at least 2 numbers, all different
    """ returns the second largest number in the list """
```

4.
```
def farthestPoint(pointList, p):
    """return the point in pointList that is
      farthest from point p """
```

"Rotating" Blocks in Quixo

- In Quixo and some other block games we need to shift blocks around by sliding them as follows:
  - Remove a block from one side of the board
  - Slide the blocks in the same row or same column to occupy vacant spots created
  - Put the removed block at other side of the board to occupy the vacant spot
- This operation is sometimes called *rotating* a list
  - Before: `['-', 'x', '-', 'o', 'o']`
  - After: `['o', '-', 'x', '-', 'o']`

Rotating blocks in Quixo

- Imagine collection of blocks stored as a list of lists of blocks
- We will be rotating rows or columns in the list of lists
- This idea can be described as rotating lists

Write and test these functions

1. ```
   def rotateFromRight(alist):
       """ Rotates alist so that the rightmost element
           appears at the left end of the list """
   ```
2. ```
   def rotateFromLeft(alist):
       """Rotates alist so that the leftmost element
       appears at the right end of the list """
   ```
3. ```
   def rotateRow(alist, side):
       """Rotates alist from the left or from the right"""
   ```
4. ```
   def rotateAnyRow(atable, row, side):
       """Rotates a row of atable from the left or from
       the right"""
   ```