

CompSci 101

Introduction to Computer Science

pop	none	hybrid
pop	position	hybrid
sort	none	mutator

Oct 13, 2016

Prof. Rodger

Announcements

- Reading and RQ 11 due next time
- APT 4 is due on Tuesday
- Today:
 - A different way to process elements
 - List comprehension
 - Coming – more ways to process data
 - Exam 1 back
 - Rodger extra office hours today 3-5:30pm

Giving Back Exam 1...

on Gradescope

- Solutions posted – regrades til Oct 21.
 - Ask for regrade on gradescope
- Try working problem you missed first
 - Then look at solution
- Once you think you understand
 - Get blank sheet of paper – try again
- Understand all solutions

Exam 1 scores

90

89 89 89 89 89 89 89 89 89 89 89 89 89 89 89

88 88 88 88 88 88 88 88 88 88 88 88 88 88 88 88 88 88 88 88

87 87 87 87 87 87 87 87 87 87 87 87 87 87 87

86 86 86 86 86 86 86 86 86 86 86 86 86 86 86 86 86

85 85

84 84 84 84 84 84 84 84 84 84 84 84 84 84 84 84 84

83 83 83 83 83 83 83 83 83 83 83 83 83 83 83 83 83 83

82 82 82 82 82 82 82 82 82 82 82 82 82 82 82 82 82 82 82 82

81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81 81

80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80

79 79 79 79 79 79 79 79 79 79 79 79 79 79 79

78 78 78 78 78 78 78 78 78 78 78 78 78 78 78

77 77 77 77 77 77 77 77 77 77 77 77 77 77 77

76 76 76 76 76 76 76 76 76 76 76 76 76 76 76

75 75 75 75 75 75 75 75 75 75 75 75 75 75 75

74 74 74 74 74 74 74 74 74 74 74 74 74 74 74

73 73 73 73 73 73 73 73 73 73 73 73 73 73 73

72 72 72 72 72 72 72 72 72 72 72 72 72 72 72

71 71 71 71 71 71 71 71 71 71 71 71 71 71 71

70 70 70 70 70 70 70 70 70 70 70 70 70 70 70

69 69 69 69 69 69 69 69 69 69 69 69 69 69 69

68 68 68 68 68 68 68 68 68 68 68 68 68 68 68

67 67 67 67 67 67 67 67 67 67 67 67 67 67 67

66 66 66 66 66 66 66 66 66 66 66 66 66 66 66

65 65 65 65 65 65 65 65 65 65 65 65 65 65 65

64 64 64 64 64 64 64 64 64 64 64 64 64 64 64

63 63 63 63 63 63 63 63 63 63 63 63 63 63 63

62 62 62 62 62 62 62 62 62 62 62 62 62 62 62

61 61 61 61 61 61 61 61 61 61 61 61 61 61 61

60 60 60 60 60 60 60 60 60 60 60 60 60 60 60

59 59 59 59 59 59 59 59 59 59 59 59 59 59 59

58 58 58 58 58 58 58 58 58 58 58 58 58 58 58

57

56 56 56 56 56 56 56 56 56 56 56 56 56 56 56

54 54 54 54 54 54 54 54 54 54 54 54 54 54 54

53 53 53 53 53 53 53 53 53 53 53 53 53 53 53

51 51 51 51 51 51 51 51 51 51 51 51 51 51 51

50

49

48

47

46

45 45 45 45 45 45 45 45 45 45 45 45 45 45 45

43

42

41 41 41 41 41 41 41 41 41 41 41 41 41 41 41

40

39 39 39 39 39 39 39 39 39 39 39 39 39 39 39

38

36

35

34

31

29

28

25

22

90

89 89 89 89 89 89 89 89 89 89 89 89 89 89

88 88 88 88 88 88 88 88 88 88 88 88 88 88 88 88

87 87 87 87 87 87 87 87 87 87 87 87 87 87

86 86 86 86 86 86 86 86 86 86 86 86 86 86 86

85 85 85 85 85 85 85 85 85 85 85 85 85 85 85 85

84 84 84 84 84 84 84 84 84 84 84 84 84 84

83 83 83 83 83 83 83 83 83 83 83 83 83 83 83

82 82 82 82 82 82 82 82 82 82 82 82 82 82 82

81 81 81 81 81 81 81 81 81 81 81 81 81 81

80 80 80 80 80 80 80 80 80 80 80 80 80 80 80

79 79 79 79 79 79 79 79 79

78 78 78 78 78 78 78

77 77 77 77 77 77 77

76 76 76 76 76 76 76 76 76

75 75 75 75 75 75 75 75

74 74 74 74

73 73 73 73 73 73 73

72 72 72 72 72

71 71 71 71 71 71 71

70 70 70

69 69 69 69

68 68 68 68 68 68 68

67 67 67 67 67 67 67

66 66 66

65 65 65 65

64 64 64 64 64

63 63 63

62 62

61 61 61 61 61

60 60 60 60 60 60

59 59

58 58 58 58

Exam 1 scores

}

Wow

}

Yes

}

ok

57

56 56 56 56

54 54

53 53

51 51

50

49

48

47

46

45 45 45

43

42

41 41

40

39 39 39

38

36

35

34

31

29

28

25

22

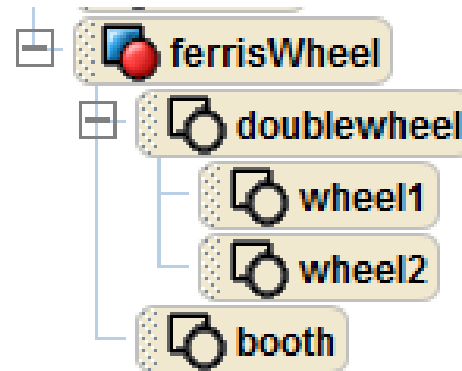
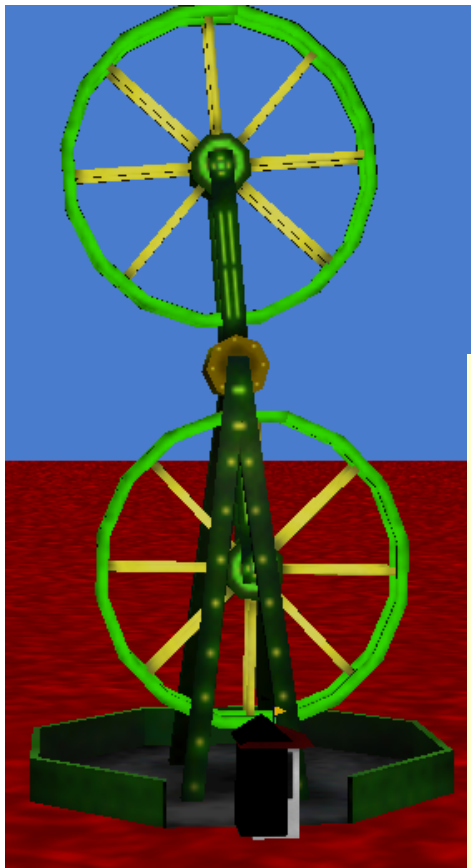
Get a tutor -
Understand when
you get help

compsci101 fall 2016



Alice programming language

alice.org, Alice version 2.4



Loop 10 times times show complicated version

ferrisWheel.doublewheel.wheel2 roll left 0.1 revolutions

Wait 2 seconds

ferrisWheel.doublewheel roll left 0.5 revolutions more...

Loop 10 times times show complicated version

ferrisWheel.doublewheel.wheel1 roll left 0.1 revolutions

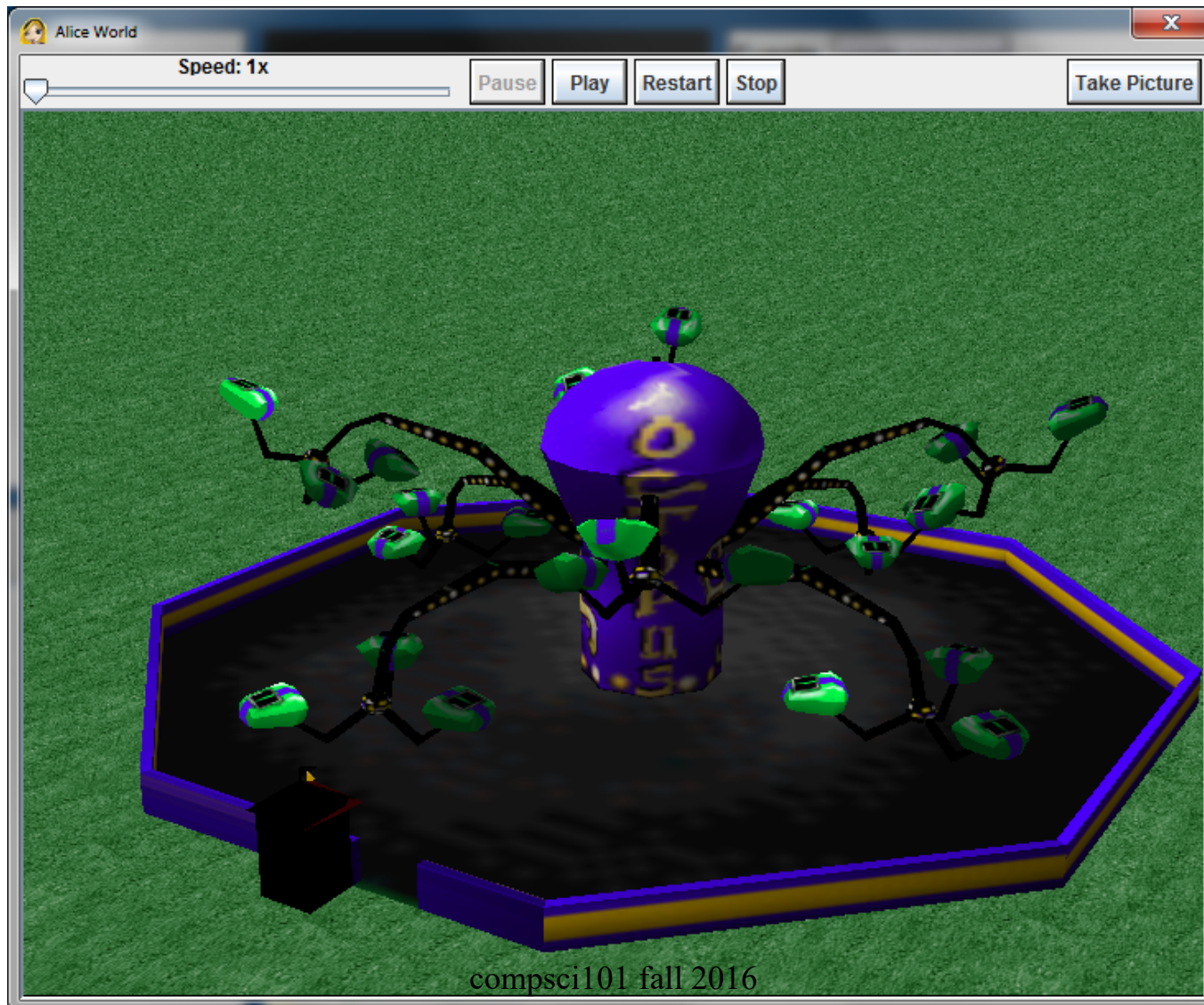
Wait 2 seconds

Nested Loop

The screenshot displays a visual programming interface with three nested levels of control structures, each indicated by a small square icon on the left:

- Outer Loop:** A light blue bar containing a "Loop" block set to "10 times" and a "show complicated version" button.
- First Nest:** A purple bar containing a "Do together" block with a single action: "ferrisWheel.doublewheel" rolling "right" for "1 revolution" with "style = abruptly" and "duration = 2 seconds".
- Second Nest:** A light blue bar containing a "Loop" block set to "2 times" and a "show complicated version" button.
- Innermost Nest:** A purple bar containing a "Do together" block with two parallel actions:
 - "ferrisWheel.doublewheel.wheel1" rolling "left" for "1 revolution" with "style = abruptly".
 - "ferrisWheel.doublewheel.wheel2" rolling "left" for "1 revolution" with "style = abruptly".

Fair Ride – Octopus



Wac-A-Mole



World.score **Obj** clicked

No variables

[-] For all World.moles ▾, one **Obj** item_from_moles at a time

[-] If clicked ▾ == item_from_moles ▾ ▾

[-] Do together

playerScore ▾ move up ▾ .2 meters ▾ duration = 0.25 seconds ▾

item_from_moles ▾ play sound World.pop2 (0:00.313) ▾ more... ▾

Else

(Do Nothing)

World's details

properties methods function

Obj moles = mole, mole2, mole3, mole4, mole5, mole6, mole7, mole8, mole9, mole10, mole11, mole12

Problem:

Find the first duplicate adjacent words in a phrase

- “Did I make make a mistake in in this?”
- Convert to list
- Answer is 2, (start counting at 0)

Bit.ly/101f16-1013-A

```
def positionDuplicate(phrase):  
    words = phrase.split()  
    if len(words) < 2:  
        return -1  
    pos = 0  
    while True:  
  
        pos = pos+1  
    return pos
```

APTs solved in a similar way

- NameGroup
- Pikachu

APTs solved in a similar way

- NameGroup
- Pikachu



Creating a list

- Given a list of numbers, create a second list of every number squared.

```
nums = [8, 3, 5, 4, 1]
```

```
sqnums = []
```

```
for v in nums:
```

```
    sqnums.append(v*v)
```

```
print sqnums
```

```
[64, 9, 25, 16, 1]
```

More on List operations

- Previous page
 - nameOfList “dot” function (parameter)
`sqnums.append(v*v)`
- See list operations on next page
- Mutator vs hybrid vs return
 - Mutator changes the list (no return value)
 - Hybrid changes list and returns value
 - Return – returns value, no change to list

List operations from book

Method	Parameters	Result	Description
append	item	mutator	Adds a new item to the end of a list
insert	position, item	mutator	Inserts a new item at the position given
pop	none	hybrid	Removes and returns the last item
pop	position	hybrid	Removes and returns the item at position
sort	none	mutator	Modifies a list to be sorted
reverse	none	mutator	Modifies a list to be in reverse order
index	item	return idx	Returns the position of first occurrence of item
count	item	return ct	Returns the number of occurrences of item
remove	item	mutator	Removes the first occurrence of item

Problem

- Remove all negative numbers from list
 $[4, -2, 5, 6, -3] \rightarrow [4, 5, 6]$
- Two ways
 - 1) return a new list with all negative numbers removed
 - 2) Modify a list to remove negative numbers

www.bit.ly/101f16-1013-2

```
def removeNegatives(numberlist):  
    answer = []  
    for num in numberlist:  
        if num >= 0:  
            answer.append(num)  
    return answer
```

somenums = [3, -1, 8, -5, -2, 6, 7]

nonegs = removeNegatives(somenums)

www.bit.ly/101f16-1013-3

```
def removeNegatives2(numberlist):  
    for x in range(len(numberlist)):  
        value = numberlist[x]  
        if value < 0:  
            numberlist.pop(x)
```

```
somenums = [3, -1, 8, -5, -2, 6, 7]  
removeNegatives2(somenums)
```

www.bit.ly/101f16-1013-4

```
def removeNegatives3(numberlist):  
    pos = 0;  
    while (True):  
        if pos >= len(numberlist):  
            break  
        value = numberlist[pos]  
        if value < 0:  
            numberlist.pop(pos)  
        pos = pos + 1
```

somenums = [3, -1, 8, -5, -2, 6, 7]

removeNegatives3(somenums)