Carnegie Mellon University
Computer Science Academy

CS1: Unit 5 Review

Complex Conditionals

Compound Conditionals

- Conditionals that use the keywords `and`, `or`, or `not` to combine simple tests into more complex tests
- Using `and` tests whether *both* are True
- Using `or` tests if *at least one* test is True
    - Python uses inclusive **or** meaning that the whole complex conditional is true if either are true or if both are true
- Using `not` tests whether a statement is false

Nested Conditionals

- Another way to deal with complex conditionals is by nesting `if` statements *inside* other `if` statements
- The conditionals inside will only be tested if the outside conditional is true
    - This also means that code in the body of the inside conditionals will only run if both the outside and inside tests are true

More Key Events

onKeyHold

- `onKeyHold` is called repeatedly, until we release the key
- You can hold down multiple keys at once, so the parameter to `onKeyHold` is keys instead of just key
- When checking if a certain key is being held, use `in`:
    - Ex. To check if the key 's' was being held: `if ('s' in keys):`
- Using compound conditionals you can check if multiple keys have been held or certain combinations
- A common use of onKeyHold is changing the location of an object
    - By checking if the location of the object is at the edge of the canvas, we can mimic wraparound and reposition the object to the other side of the canvas
- Note: the autograder uses `onKeyHolds` but this is purely for test cases and should not be used in your code