



Boolean Expressions and if

2/1/2008





Opening Discussion

- Let's look at solutions to the interclass problem.
- Minute essay questions.
 - ◆ Turning numbers into strings.
 - ◆ Making a function return a string.
 - ◆ Using variable: when, where and how?
 - ◆ Moving properties/methods/functions from the world to a different world.
 - ◆ Glossary of terms.





Problem

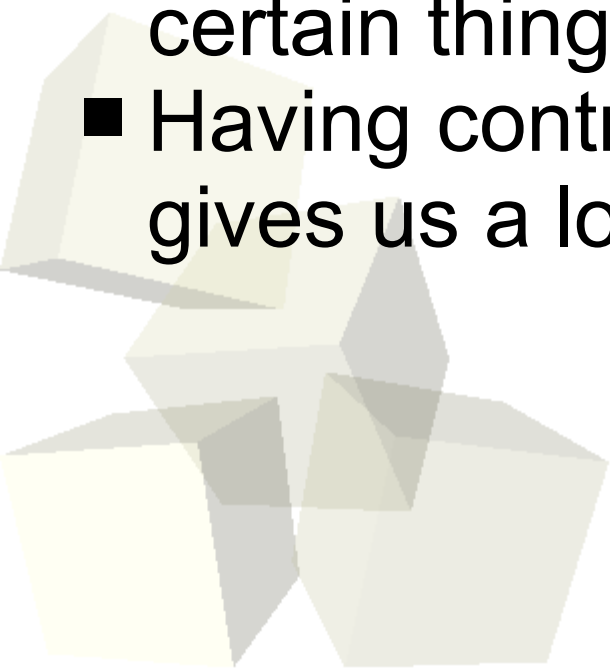
- I want the mummy to attach the character closest to him first.
- Right now the mummy always attacks the penguin first and the skeleton second. I want him to attack the closer one first and the other one second.





Conditional Execution

- So far all of our programs have been fairly simple in that the same lines of code happen in the same order every time we run the program.
- Changes in user input or initial conditions might have changed certain details, but basically the same thing happened every time we ran the code.
- Real programs need to have the ability to do certain things only in certain situations.
- Having control over what pieces of code execute gives us a lot more power.





Boolean Type

- The Boolean type is an extremely simple type. It can hold one of two values. It is either true or it is false.
- This type is simple, but it is the foundation of logic.
- We have the ability to specify Boolean as the type of our variables, parameters, properties, or function returns.





Boolean Functions

- Alice objects have many functions that return Boolean values. They all start with the word “is”.





Boolean Expressions

- We can also build complex Boolean expressions using the relational operators and Boolean operators provided in the functions under the World.
- The relational ones are fairly clear other than == for equality and != for inequality.
- The Boolean operators are:
 - ◆ ! - not
 - ◆ && - short-circuit and
 - ◆ || - short-circuit or (not exclusive)





The if Statement

- The first place we will use Boolean values and Boolean expressions is in the if statement.
- This allows us to check if something is true or not and execute different pieces of code.
- In Alice the if statement requires we provide a boolean expression. Then there are two blocks of code. The first happens if the expression is true and the second happens if the expression is false.





- Most of the time when you say “or” in English you mean the exclusive form where we don't accept both being true. How would you make an exclusive or using the boolean operations we have in Alice (&&, ||, !)?
- Interclass Problem – Take your animated object and have it check the distance to another object. If it is more than 3 meters away have it turn toward that object and take a step. Do this three times so the first object walks toward the second unless it gets close in to it.