

Alice Programming:

Variables, Functions, Math, and Strings

Objectives

- To understand:
 - what the many variables are and when and how they are used
 - what three things are specified when creating a variable
 - how a function differs from a method
 - where an object's functions are listed
 - what primitive functions are
 - what operators are and how are they used
 - how math expressions are used in programming
 - what a collision is and ways to avoid them
 - what a string and string concatenation are

3-2

Today's Agenda

1. Variables
2. Using Functions
3. Creating Math Expressions
4. Working with Strings and Text

3-3

Variables

- Programs may need to store data when running and it stores that data in a variable
- A **variable** is a named storage location in the computer's memory
- The following types of data (or variables) can be stored:
 - local variables
 - parameter variables
 - class-level variables
 - world-level variables

3-1

3-4

Variable Types

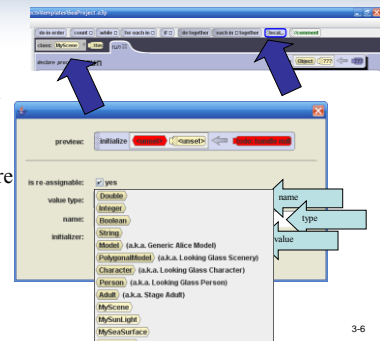
- Local variables**
 - belongs to a specific method
 - used only by the instructions that the variable belongs to
 - when a method stops, the variables cease to exist
- Parameter variables**
 - used to hold an argument that is passed to a called method
- Class-level variables**
 - variable that belongs to a specific object
- World-level variables**
 - variable that belongs to the world

3-1

3-5

Creating Local Variables

- Local variables belong to a specific method (such as *MyScene*)
- Variables require 3 things:
 - name
 - type
 - initial value



3-1

3-6

Names and Types and Initial Values

- Variable Names
 - Must be unique *within* the method
 - Should be meaningful and reflect the variable's purpose
 - Named using camelCase
- Variable Types
 - Numbers, Boolean, Objects, Other (such as String, Color, or Sound)
- Initial Value
 - Value *initially* stored in the variable

Variables

Tile appearance shows the type of variable



Creating the tile is called *variable declaration* where you *initialize* the variable



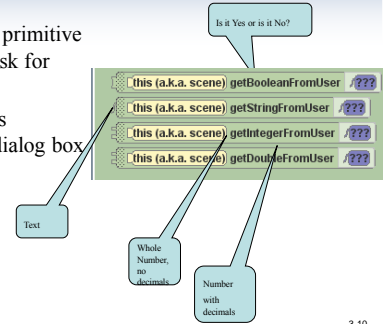
Variable Assignment

- Variables have an initial value
- Initial value held until a different value is assigned
- New values can be assigned while the method is running
 - Set instructions* can be created to store (or *set*) different values
 - Request to set a value occurs when variable tile dropped into Methods Editor



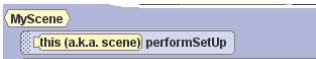
Asking the User for Input

- World has four primitive functions that ask for user input
- Each function is displayed in a dialog box



Initial Setup...Places Everyone!!!

- Some characters may need posing to begin
- Some objects may need to be a set distance from another object
- Some scenes may need special lighting or other properties
- This is handled through a *performSetup* procedure that is a procedure for the scene object



Initial Setup

- Set the characters in position, using the exact same tiles
- Order doesn't matter as this all happens when the Director (you) calls "Places Everyone"
 - The curtains open and MyScene runs
- Right-click on *performSetup* tile and choose EDIT
- Do NOT remove tiles that are already here



Opacity

- **Opacity** is a measure of how "see-through" an image or an object is.
 - ☞ The less opaque an object is, the more see-through it is.
 - ☞ Opacity of 100%, cannot see through the object
 - ☞ Opacity of 0%, object is transparent (like clear glass)
- In Alice, an object with an opacity of 0% is invisible.

Changing the Opacity



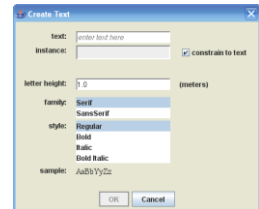
- ☞ A change in opacity can be used to simulate real world conditions.
- ☞ Example
 - A fish swimming away from the camera should fade away because water blurs our vision of distant objects.

Demo

- ☞ As opacity is gradually decreased, the fish becomes less visible.
 - At an opacity of 0%, the object is still in the world (can still see it listed in the object tree) but is effectively invisible.

Demo

- ☞ **Create Text...** can be used to display a title screen for the animation.
 - A 2-second *wait* gives the user time to read the billboard.
 - Changing the *opacity* property to 0% makes the title screen disappear.



Demo

- ☞ Enter the title in the text field
- ☞ FAMILY:
 - Serif
 - SansSerif



Practical Uses of Invisible Objects

- ☞ An invisible object is sometimes useful as a stationary marker that creates
 - a target for a *move to* instruction
 - an external reference point for object rotational motion