


1 Programming

Peeking into Computer Science



© Jalal Kawash 2010

1

- Mandatory: Chapter 5 – Sections 5.1 to 5.4
- Download Alice 2.2 from www.alice.org
 - JT: I downloaded version 2.2:
 - http://www.alice.org/index.php?page=downloads/alice2.2_archive
- Jalal's resources:
 - "How to" movies and example programs available at:
<http://pages.cpsc.ucalgary.ca/~kawash/peeking/alice-how-to.html>
- JT's resources:
 www.cpsc.ucalgary.ca/~tamj/203/extras/alice

Reading Assignment

© Jalal Kawash 2010

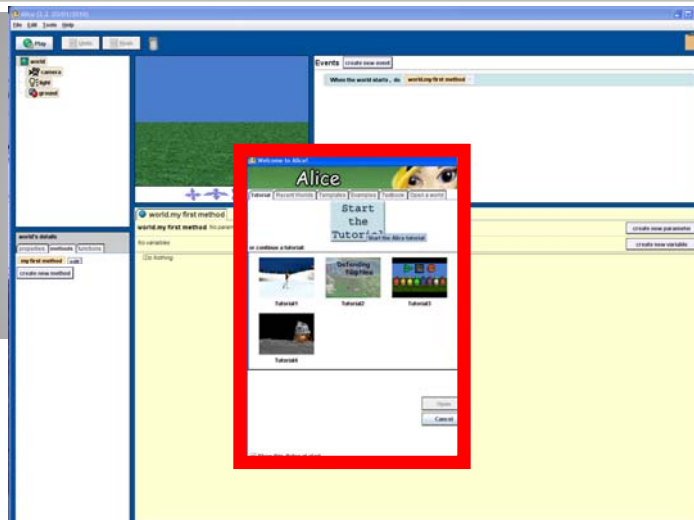
2

- A detailed step-by-step getting started guide
 - <http://www.cs.utexas.edu/~scottm/firstbytes/AliceProgrammingIntroFindingNemoCharades.pdf>
- Alice tutorials
 - <http://www.dickbaldwin.com/tocalice.htm>
- Alice training videos
 - <http://www.vtc.com/products/Programming-With-Alice-Tutorials.htm>



JT's Extra: Additional Resources

© Jalal Kawash 2010



Alice Tutorials To Complete (In Tutorial)

© Jalal Kawash 2010

- Practice things yourself.
 - "I wish he [JT] would help us more by giving us code [parts of a computer program] that can be directly used in the assignment."



- How Computer Science works: You get better by doing things for yourself (this is a 'hands-on' field of study and work).

Similar to getting fit: you can't just watch



You have to do it yourself



JT: How To Study/Prepare For This Section?

Peeking into Computer Science

© Jalal Kawash 2010

5

- Write (lots of) programs.
 - At the *very least* attempt the assignment and quiz.
 - Try to do some additional practice work (some examples may be given in class).
 - Write lots of little 'test' programs to help you understand and apply the concepts being taught.
 - Example: branching/if was covered in class, write a program that employs branching.
- Read (lots of) programs: 'tracing'
 - Reading through programs that other people have written, and executing it 'by hand' (getting the answer on your own without running the program) in order to understand how and why it works the way that it does.



JT: What Can You Do To Practice?

Peeking into Computer Science

© Jalal Kawash 2010

6

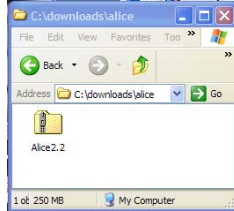
- If you find concepts unclear trying to understand them on your own can be beneficial (because this is a 'hands on' field).
 - Looking at online resources:
 - Remember when looking for academic resources, just like resources for other web searches, not all websites are good sources.
 - Start with more reputable sources e.g., www.alice.org and the ones that I may have recommended.
- If you are still unclear on concepts then make sure that you ask for help.
 - Don't wait too long to do this because latter concepts may strongly depend on the understanding of earlier concepts.

JT: Asking Questions

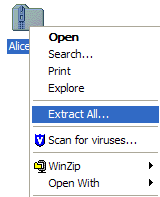
- Don't cram the material just before the exam, instead you should be studying the concepts as you learn them throughout the term.
- It's important to work through and understand concepts *before* you start the assignment. If you try to learn a new concept and work out a solution for the assignment at the same time then you may become overwhelmed.
- Don't start assignments the night (or day!) that they are due, they may take more time than you first thought so start as soon as possible.

JT: Start Studying/Working On The Assignment As Early As Possible

- Download version: 2.2
- What you get: A compressed (zip) file.



- This file needs to be uncompressed.



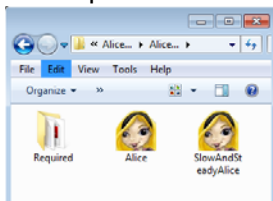
JT's Extra: Setting Up Alice

Peeking into Computer Science

© Jalal Kawash 2010

9

- Alice (2.x) doesn't need to be installed on your computer after it's been uncompressed.
- Just find the folder containing the executable program:
 - "Alice" or "SlowAndSteady" (recommended)
 - These programs can be found in the folder where you uncompressed the Alice files (Alice2.2)



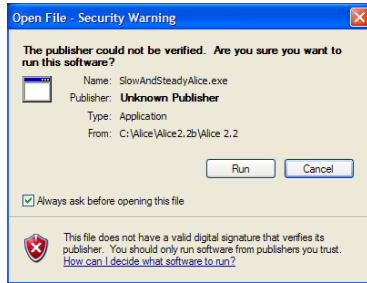
JT's Extra: Starting Alice (2)

Peeking into Computer Science

© Jalal Kawash 2010

10

- If you see this message you should normally pay attention!



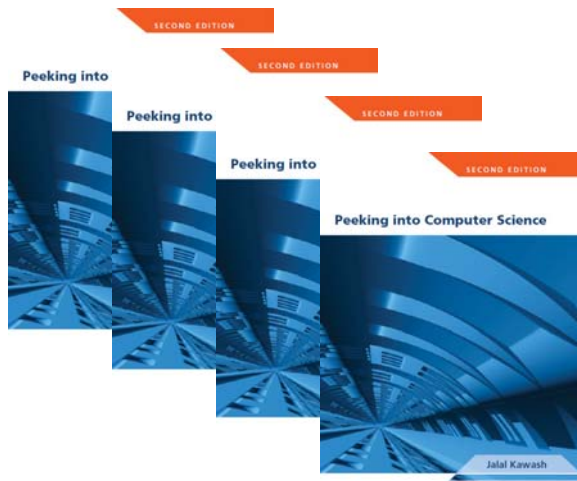
- If it's for a program that you trust then it's safe to proceed otherwise you should proceed with caution (probably select 'cancel').
 - If you downloaded Alice (from the correct website) then you can just ignore it.



JT's Extra: Scary Message????!!!

Peeking into Computer Science

© Jalal Kawash 2010



Programs

Communicating with Computers

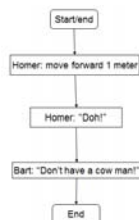
At the end of this section, you will be able to:

1. Differentiate between high-level and low-level programs
2. Differentiate between the two types of translators

Objectives

- Algorithms can be specified in different ways

Graphical flowchart



Pseudo code (sort of program code)

Repeat the following steps:

1. If $RS = W$ and $FS = S$, then **F**
2. If $FS = W$, then **L**
3. If $RS = S$ and $FS = S$, then **R** and **F**

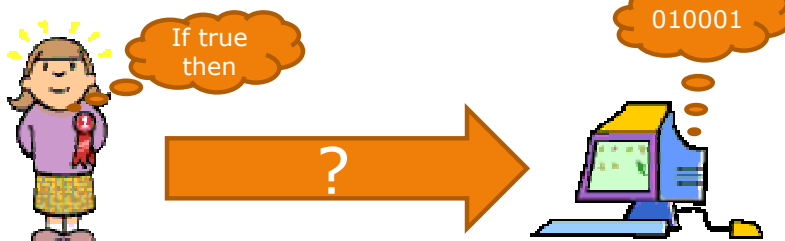
Algorithms

- JT: For this part of the course we focus on specifying algorithms using a programming language (Alice)



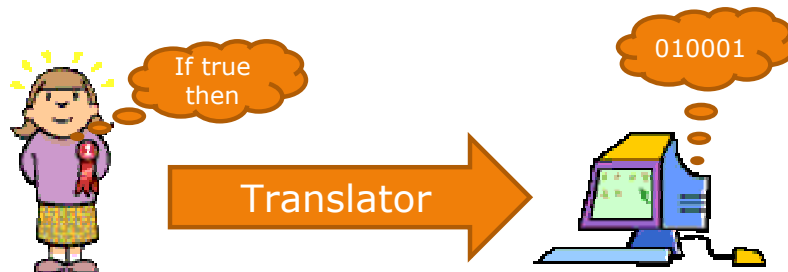
Algorithms (2)

- Computers understand 0s and 1s
 - Machine language
 - Low Level (LL) language
- Programmers use High Level (HL) languages
 - Easier to understand and work with



Programming Levels

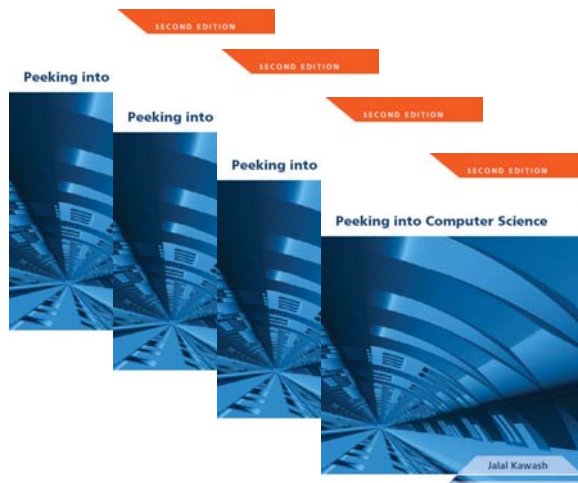
- Translator: a program that translates HL to LL code
- Two types: Compilers & Interpreters



Translation

- Compilers: translate the whole HL program to LL program
- Interpreters: translate the HL program to LL program, one instruction at a time

Translators



Objects and Classes

Object-Oriented Programming

19

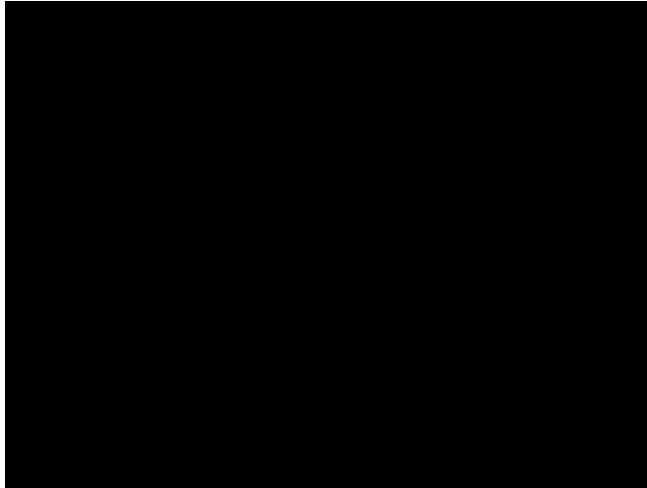
At the end of this section, you will be able to:

1. Understand what objects are
2. Understand properties and behavior of objects
3. Understand what classes are
4. Understand how objects and classes are related
5. Understand object composition

Objectives

Peeking into Computer Science

© Jalal Kawash 2010



An Example Alice Movie

- Alice is an Object-Oriented language
- An Alice program is called a *world*
- Everything in the scene is an *object*:
 - Cow, Trex, mill, grass, etc...

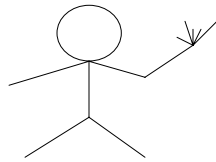


Alice Concepts

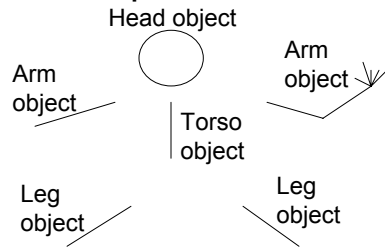
- JT: objects are the physical things that can be 'seen' and 'grasped'.



- Objects can also be composed of other objects.



'Person' object



JT's Extra: More On Objects

Peeking into Computer Science

© Jalal Kawash 2010

- An object is similar to an entity in Databases
 - It has *properties* (attributes)
 - In addition, it has *behaviour*
- Cow objects have at least the properties:
 - Size and color
- Different objects may have different values for properties



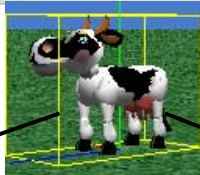
Objects

Peeking into Computer Science

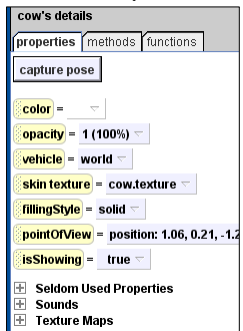
© Jalal Kawash 2010

- Objects have behaviour
- A cow can talk, walk, and eat
- The behaviour is specified as a collection of *methods*
- A method is a small program
 - Allows the object to do something specific

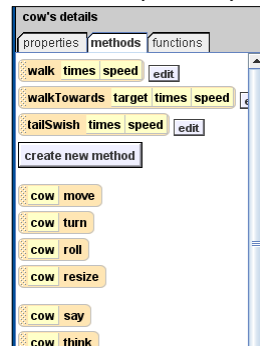
Object Behaviour



Attributes (information)

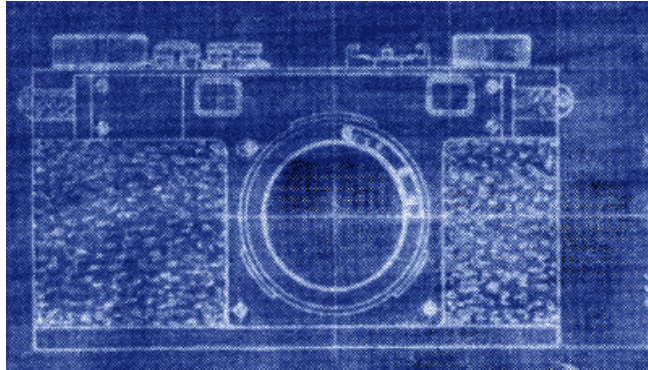


Behaviors (actions)



JT's Extra: Attributes Vs. Behaviors

- It is a general template.
- Specifies the methods and attributes of objects which are examples/instances of the class.

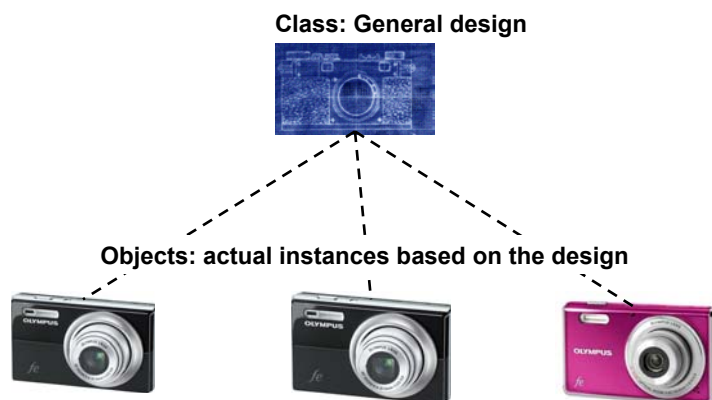


JT's Extra: A Class Is Like A Blueprint

Peeking into Computer Science

© Jalal Kawash 2010

- A class is used as a design guide from which objects are created.
- Objects are instances of or examples of a class.



JT's Extra: Objects Vs. Classes

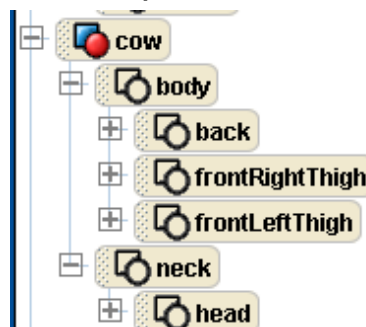
Peeking into Computer Science

© Jalal Kawash 2010

- A class is similar to an entity **type** in Databases
 - It is a blueprint for objects
- Objects are called *instances* of classes
- Creating an object *instantiates* the object
 - JT: Instantiation is the creation of an actual instance of an object ("instant") from the description provided in the class.

Classes

- An object can be made from other objects
 - JT: as described earlier "decomposed person" example
- The cow object has two component objects:
 - Neck and body
 - Further composed of other objects

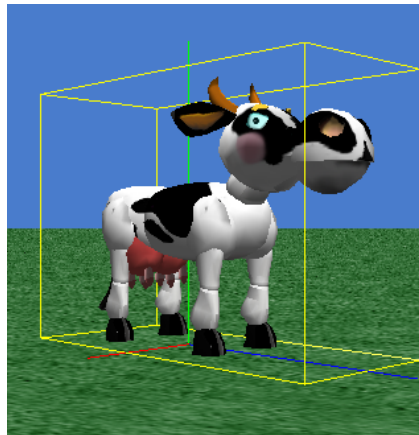


Composition

- An object-oriented program is a collection of objects
 - An Alice world is an OO program
- The objects exhibit a certain behaviour through *executing* their methods
 - *Calling* their methods
- *cow.turn, trex.turn, trex.walk, windmill.blades.roll*

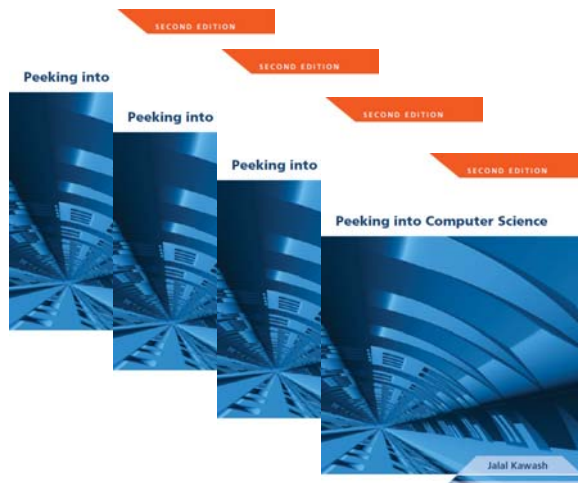


Object-Oriented Programs



Directions, Center Point, and Bounding Box





Meet Alice

The basics

33

At the end of this section, you will be able to:

1. Identify the 5 basic components of an Alice window
2. Understand the functions of each such component
3. Go through the steps to create an Alice program

Objectives

Peeking into Computer Science

© Jalal Kawash 2010

The screenshot shows the Alice World Window interface. Red circles highlight several key components:

- Object tree:** A hierarchical list of objects in the world, including 'world', 'camera', 'light', 'ground', 'windmill', 'cow', 'trex', and 'tall'.
- World view:** A 3D perspective view of the virtual world, showing a landscape with a windmill and a dinosaur.
- Event editor:** A panel for defining events, such as 'When the world starts, do world.my first method'.
- Details window:** A panel for editing the properties and methods of a selected object, currently showing 'world.my first method'.
- Method editor:** A panel for editing the code of a method, showing a sequence of actions like 'turn left', 'say', 'play sound', 'move forward', etc.

Alice World Window

Peeking into Computer Science © Jalal Kawash 2010

The screenshot shows the Alice program interface. A 'Welcome to Alice' dialog box is open, displaying a 'Templates' window with various pre-made worlds:

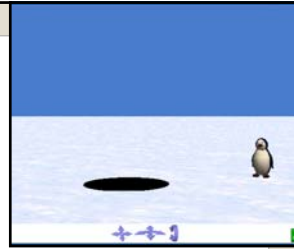
- Templates:** A grid of world templates including 'dirt', 'grass', 'sand', 'snow', 'space', and 'water'.

 The background shows the Alice World Window interface with the 'Object tree' and 'World view' visible.

Creating an Alice Program

Peeking into Computer Science © Jalal Kawash 2010

- Your computer program will be written for a virtual world.
- There are different worlds that can be chosen but generally it's just the appearance that differs.
- Your view of the world can be changed using the controls.
 1. Up, down, left, right
 2. Forward, backward, left, right
 3. Tilt forward and backward
- *(JT's note: don't get too carried away with the camera controls)*



Perspective controls

– 1. World (JT's Extra)



- The world contains objects that are shown in tree form.
- Although your world will consist of three objects by default (camera, light, ground) it's the extra objects that you add to this world that will be of interest most of the time.
 - (In the example to the right the extra objects: penguin, circle).



– 2. Object Tree (JT's Extra)



– Adding Objects (JT's Extra)

Peeking into Computer Science © Jalal Kawash 2010

- Different events can occur in the world

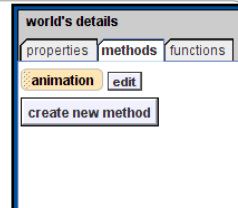
create new event
When the world starts
When a key is typed
When the mouse is clicked on something
While something is true
When a variable changes
Let the mouse move <objects>
Let the arrow keys move <subject>
Let the mouse move the camera
Let the mouse orient the camera

- The event that we'll focus on for now is when the world starts (simulation begins running).

– 3. World Events (JT's Extra)

Peeking into Computer Science © Jalal Kawash 2010

- Used to see the details of the virtual world (or objects in the world).
- Details:
 - Physical characteristics (“properties” in Alice).
 - Actions (“methods” in Alice).
 - Actions that perform an action and *generates a value* (“functions” in Alice).



4. World Details (JT's Extra)



- Properties provide information about an object

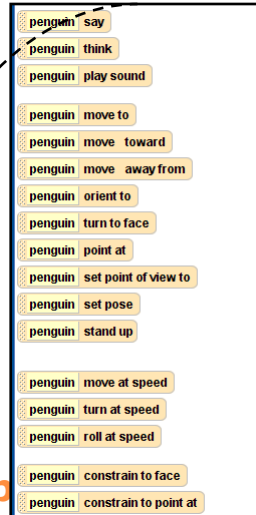


- Many (most) properties can be changed.

Properties Of An Example Object: Penguin (JT's Extra)



- Methods are the actions that an object is capable of carrying out.

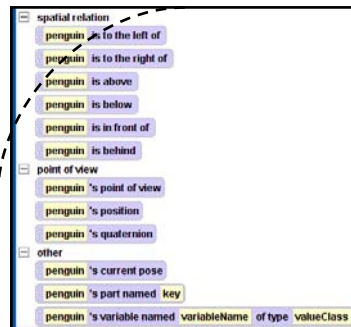
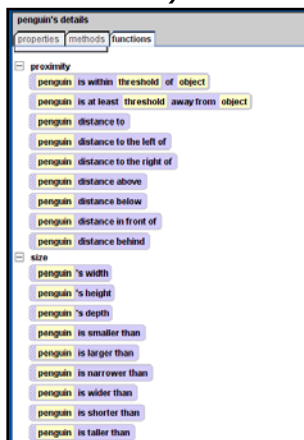


Methods Of An Example Object: Penguin (JT's Extra)

Peeking into Computer Science

© Jalal Kawash 2010

- Functions are typically asking a question (logical statement).

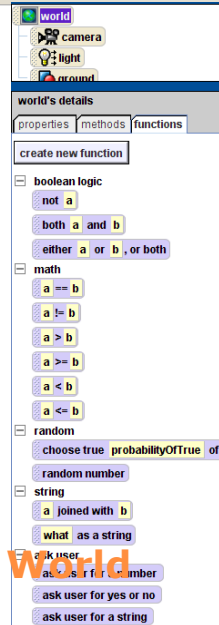


Functions Of An Example Object: Penguin (JT's Extra)

Peeking into Computer Science

© Jalal Kawash 2010

- Logical operations (“Boolean logic”, “math”)
- Generating random numbers
- Prompting the user of the program for information (“ask user”)

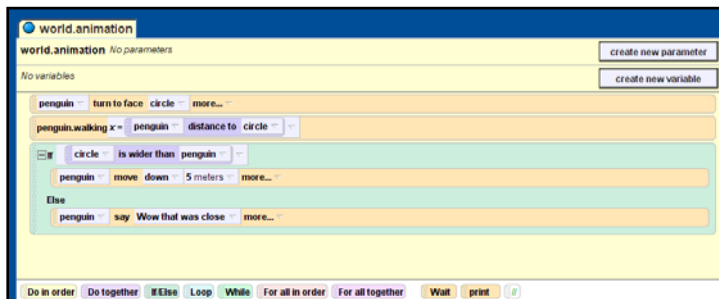


Some Useful Functions Of The World (JT's Extra)

Peeking into Computer Science

© Jalal Kawash 2010

- A computer program consists of a series of instructions
 - (e.g., save document, print document, spell check etc.)
- This is where you use the methods, functions and the other parts of Alice to 'program' (create) those instructions.



5. Event (Method) Editor (JT's Extra)

Peeking into Computer Science

© Jalal Kawash 2010

- Must have a story line before you start programming:

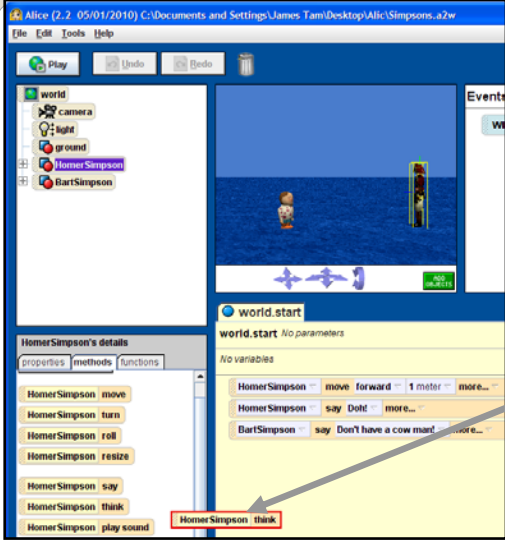
1. In an open field with a windmill in the background,
2. a trex approaches a cow from behind;
3. the cow turns towards the trex, says "Trex, I will show you", and moos;
4. the trex, says "Oh please no", turns, and runs away, until she disappears from the scene;
5. the cow says "where does she think she is", moos, continues "Jurassic Park?", and moos.

Story Lines

- Basic instructions are provided in a linear fashion (one after another).



JT's Extra: Programs Are Written In A Narrative Story Form

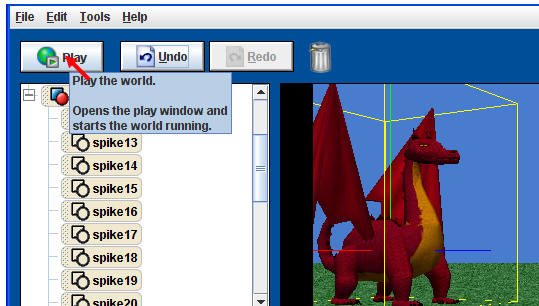


Dragging an instruction from the available list into the editor.

JT's Extra: Instructions Are Given To The Program Visually

Peeking into Computer Science © Jalal Kawash 2010

- This is when the instructions in your program are executed.



JT's Extra: Running Your Program

Peeking into Computer Science © Jalal Kawash 2010

- This method is executed when you run an Alice world
- Call the methods you want to run on objects from this method
- This is your *main* program
- Rename to a sensible name e.g., "start", "main"



JT's Extra: World.my first method