Programming with Alice

Week 4 Events

Objectives
- Explore the different types of events that can be processed in an Alice world.
- Distinguish between events that fire once and those that fire repeatedly.
- Examine the difference between loop processing and event processing.
- Create events that respond immediately to input from the keyboard and mouse.
- Use events to monitor conditions and changes in the values of variables.

Chapter 4 PowerPoint
Source Code for CH 4

4.1 Introduction to Event Processing
Events can be generated by the keyboard or the mouse in Alice. We can tell our programs to pay attention to certain events and ignore others.

4.2 World Events
Windmill
TryThis: 1–2

### World Events

<table>
<thead>
<tr>
<th>When the world starts</th>
<th>When the world is running</th>
</tr>
</thead>
</table>

Default when the world starts.

<table>
<thead>
<tr>
<th>Events</th>
<th>create new event</th>
</tr>
</thead>
</table>

Change to while enables continuous activity

<table>
<thead>
<tr>
<th>Events</th>
<th>create new event</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>When the world starts, do</th>
<th>world.my first method</th>
</tr>
</thead>
<tbody>
<tr>
<td>delete</td>
<td></td>
</tr>
<tr>
<td>change to disable</td>
<td>While the world is running</td>
</tr>
</tbody>
</table>
While the world is running

Continuous motion of the Windmill Blades during the run of the world program.

4.3 Keyboard Events

Rockette

TryThis: 3–6:

Changing the default when to a while
Basic form of the *while* key event

Basic form of allowing an *arrow to move an object*.

Rockette: Lisa.kick method

Rockette, Rocker, Windmill

TryThis: 7–11:

### 4.4 Mouse Events

This is from *The Rocker world*.

<table>
<thead>
<tr>
<th>Mouse Events</th>
</tr>
</thead>
<tbody>
<tr>
<td>When the mouse is clicked on something</td>
</tr>
<tr>
<td>While the mouse is pressed on something</td>
</tr>
<tr>
<td>Let the mouse move the camera</td>
</tr>
<tr>
<td>Let the mouse orient the camera</td>
</tr>
<tr>
<td>Let the mouse move objects</td>
</tr>
</tbody>
</table>
Let the mouse move the camera.

Events
- create new event
- Let mouse move the camera

Let the mouse orient the camera.

Events
- create new event
- Let mouse orient the camera

Moving several objects at a time.

Events
- create new event
- Let mouse move Any Object

4.5 Condition Events
Concert, Cow
TryThis: 12–15:

**Condition Events**
- When a variable changes
- While something is true
- When something becomes true

Events
- create new event
- When mouse is clicked on left button, button do
  - If left button.depressed is true
    - Begin Nothing
    - During left button, playMusic
    - End Nothing
  - Else
    - left button.depressButton

While left button.depressed is true
- Begin Nothing
- During left button, playMusic
- End Nothing
This is from the Cow world's my first method.

world.my first method No parameters

No variables

// Mooing Cow.a2w

Wait 0.5 seconds

While world.secondsSinceStart < 11

Do together
cow.tailSwish times = 2 speed = 2
time sign set text to world.secondsSinceStart as a string
increment world.secondsSinceStart by 1
Wait 1 second

world.mooTime set value to true

Do together
cow.headTurn
cow.openJaw
4.6 More to Explore

Object Groups– helps organize the object tree. You cannot manipulate a group as a single entity. We have to wait a bit to learn how to do this….worth waiting for!

Dummy Objects–helps setting predefined locations in your world where you’d like object and the camera to move. Order is important: add a dummy object to a location, then move the camera (or another object) to the dummy object. Appendix A will show you how to use the *drop dummy at camera* button.

Input Dialogs–User input at last!

Print–Great place to echo information to the screen while the program is running like the value of variables.