

5 Event Handling

Interactive Programming

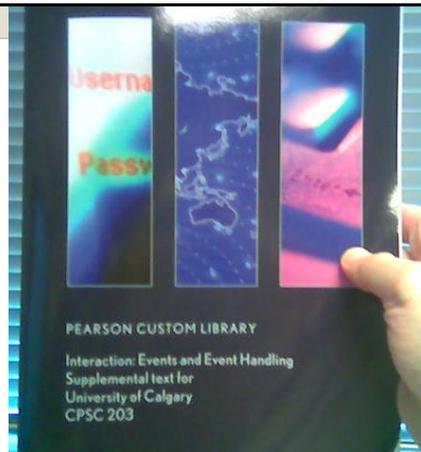
- Interaction: Events and Event Handling, Supplemental Text for CPSC 203
- Distributed this term with new textbook copies

- If you bought a used copy, then read the lab manual:

<http://pages.cpsc.ucalgary.ca/~kawash/peeking/labManual.html>

- JT's examples:

http://pages.cpsc.ucalgary.ca/~tamj/203/extras/alice/examples_part5.htm



Suggested Reading

By the end of the section, the student will:

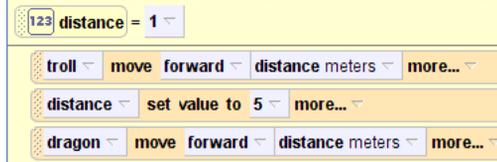
1. Understand events and event handling
2. Create and use 9 events and even handlers in Alice
3. Start working on interactive games

Objectives

- Programs seen so far do not interact with the user
- Interactive programs: react to user events
 - Such as: moving the mouse, clicking, or pressing a key on the keyboard
- Interactive games are one example

Interactive Programming

Simple program: pure linear sequence



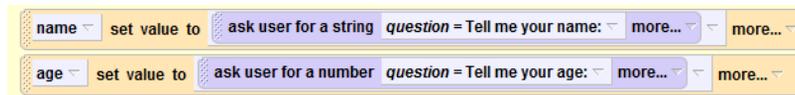
Program with loops: not purely linear but the timing of execution can be directly predicted



JT (Until Now): Program control is largely determined by the program through a series of sequential statements.

- The user can only interact with the program at places that are specified by the program (e.g., when an input statement is encountered).

Examples



JT: Traditional Software

Example 14

- Program control can also be sequential



```

Loop 100 times times show complicated version
┌ Do together
│ nerd_sheldon_reflection.rightArm roll left 0.25 revolutions more...
│ nerd_sheldon_reflection.leftArm roll right 0.25 revolutions more...
│ nerd_sheldon say I am sooooo cool that it's painful! more...
│ nerd_sheldon_reflection.rightArm roll right 0.25 revolutions more...
│ nerd_sheldon_reflection.leftArm roll left 0.25 revolutions more...
└

```

JT: Event-Driven Software

Regular code instructions

```

Loop 100 times times show complicated version
┌ Do together
│ nerd_sheldon_reflection.rightArm roll left 0.25 revolutions more...
│ nerd_sheldon_reflection.leftArm roll right 0.25 revolutions more...
│ nerd_sheldon say I am sooooo cool that it's painful! more...
│ nerd_sheldon_reflection.rightArm roll right 0.25 revolutions more...
│ nerd_sheldon_reflection.leftArm roll left 0.25 revolutions more...
└

```

Always runs when
program runs



Event:
When???

```

world.startOfTheWorld No parameters
No variables
ground move up 100 meters more...

```

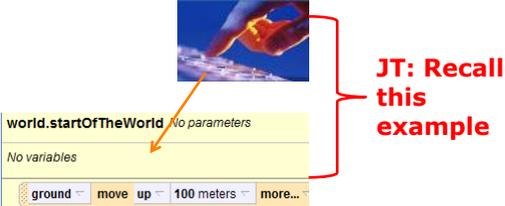
Only runs when event occurs
as program is running

JT: Event-Driven Software Can Include Code That Occurs Only When Event Occurs

- When the world starts
- When a key is typed
- When the mouse is clicked
- Let the mouse move an object
- Let the arrow keys move an object
- Let the mouse move the camera
- Let the mouse orient the camera
- While something is true
- When a variable changes

Event Types

- When an event takes place, a method is executed

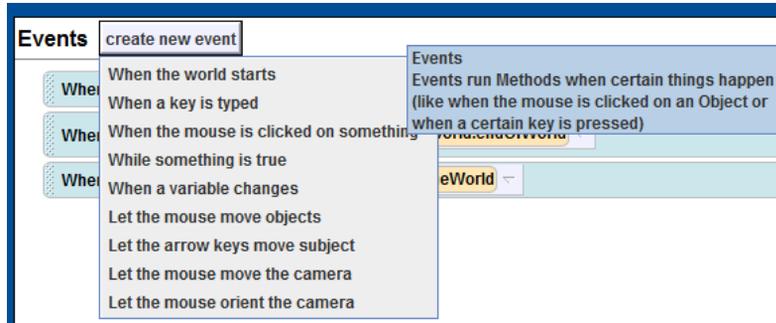


JT: Recall this example

Method 'startOfTheWorld' is an example of an event handler

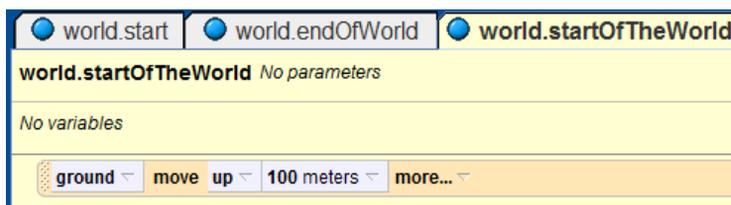
Event Handling

1. Identify the event



JT: Steps For Creating An Event Handler

2. Create a handler for it (define a method that runs when the event occurs)



JT: Steps For Creating An Event Handler

3. Register the handler for this event

When **any key** is typed, do **world.startOfTheWorld**

JT: Steps For Creating An Event Handler

- Wait until an event happens
- Every time it happens, a handler is performed
- E.g. when a mouse is clicked



world.startOfTheWorld No parameters
 No variables
 ground move up 100 meters more...

When an event happens

- Example 15: (up/down) arrows

Events

When the world starts, do world.start

When  is clicked on coach, do coach.coachReactsClick

When  is typed, do coach resize 2 more...

When  is typed, do coach resize 0.5 more...

JT: Specific Events May Be Tied To Handlers

- Example 16:

Example 16

Hates
head
petting

Likes
body
petting

Events

When the world starts, do world.start

When  is clicked on cat.head, do cat.pattedPoorly

When  is clicked on cat.frontLeftLeg, do cat.pattedGood

When  is clicked on cat.frontRightLeg, do cat.pattedGood

When  is clicked on cat.backLeftLeg, do cat.pattedGood

When  is clicked on cat.backRightLeg, do cat.pattedGood

**JT: Events Specific To Objects,
Random Reactions**

The image shows a Scratch script for a method named "catPettedPoorly". The script starts with a "play sound" block for "cat.angry 04 (0:02.016)". This is followed by a "set value to" block for "cat.direction" using a "random number" block with a minimum of 1 and a maximum of 5, and "integerOnly = true". Then, there are three conditional "if" blocks based on "cat.direction" being 1, 2, or 3. Each "if" block contains "move forward" and "turn" blocks. The "if" block for direction 1 has a "Do Nothing" block in the "Else" section. The "if" block for direction 2 has a "Do Nothing" block in the "Else" section. The "if" block for direction 3 has a "Do Nothing" block in the "Else" section. There is also a partially visible "if" block for direction 4.

Play angry cat sound

Cat's random reactions to event

JT: Method "catPettedPoorly"

The image shows a Scratch script for a method named "catPettedGood". It begins with a comment: "// Cat purr from: <http://pictures-of-cats.org/Cat-Sounds-WAV-WMA-MP3>". The script then plays the sound "cat.purr (0:02.681)". This is followed by a "Do together" block containing "play sound" for "cat.cat (0:00.586)" and "roll left" by "0.5 revolutions" with a "duration" of "3 seconds". Next is a "Loop" block set to "5 times" containing "roll left" by "0.1 revolutions" and "roll right" by "0.1 revolutions", both with a "duration" of "0.25 seconds". This is followed by another "Do together" block with "play sound" for "cat.cat (0:00.596)" and "roll right" by "1 revolution" with a "duration" of "3 seconds". Another "Loop" block set to "5 times" contains "roll left" by "0.1 revolutions" and "roll right" by "0.1 revolutions", both with a "duration" of "0.25 seconds". The script ends with a "roll right" by "0.5 revolutions".

Sounds of contented cat

Cat's happy reactions

JT: Method "catPettedGood"

- When the world starts, "world.my first method" is executed
- You can change it

When the world starts

- In a world with an monkey object
- Create a simple method of monkey, dance:



When the world starts

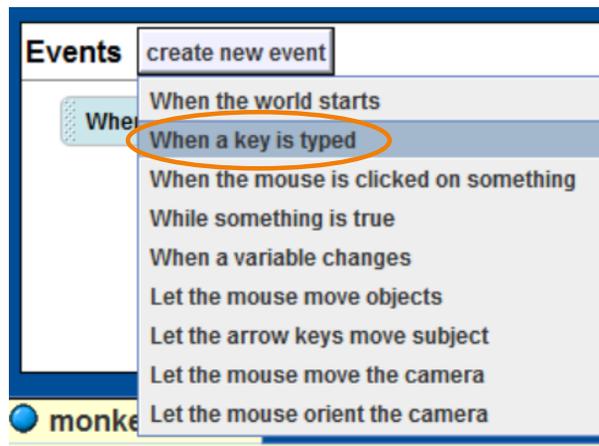
- Register the new method for the event



- When the world starts, monkey.dance will be executed instead of world.my first method

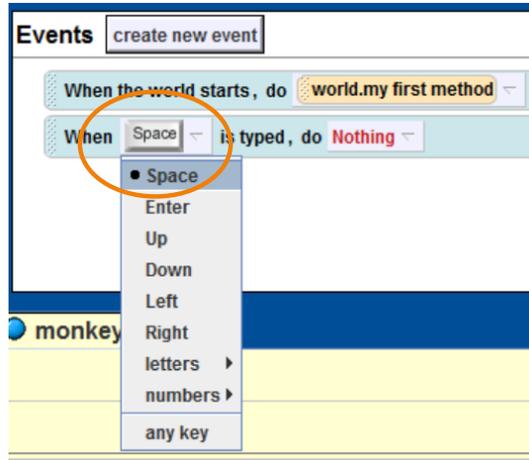
When the world starts

- Create a new event



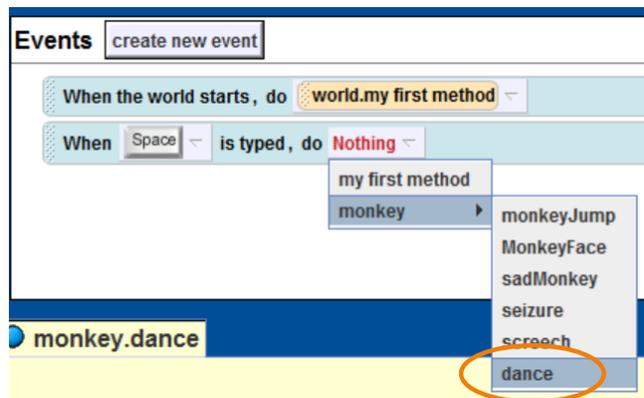
When a key is typed

- Choose the key to be typed



When a key is typed

- Choose handler method



When a key is typed

- Every time the space key is typed, the monkey dances



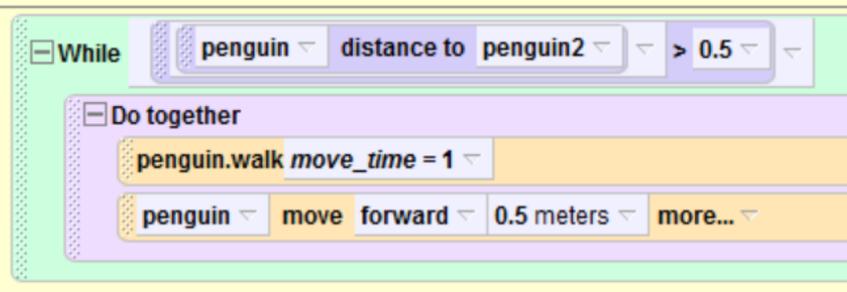
When a key is typed

- A world with two penguins facing each other



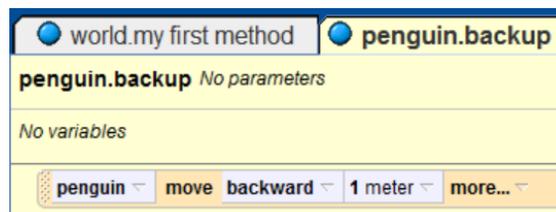
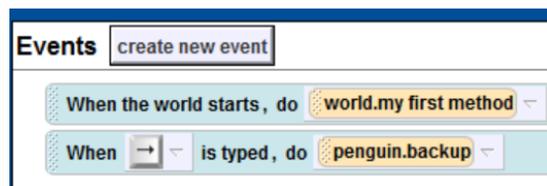
Example

- White penguin moves towards pink penguin until too close



Example

- Holding the white penguin back: event and its handler

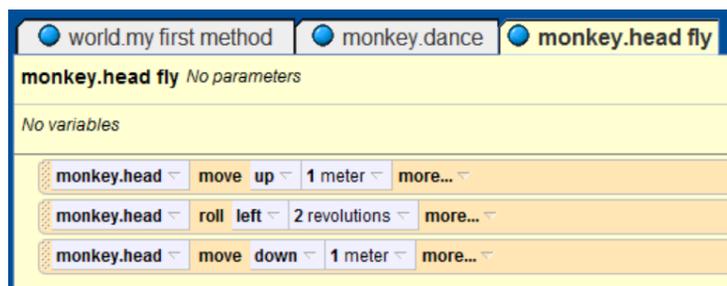


Example

- Create an Alice world and add one object to it
- Write handlers so that:
 - UP arrow key: moves the object up ½ meters
 - DOWN arrow key: moves the object down ½ meters
 - LEFT arrow key: moves the object left ½ meters
 - RIGHT arrow key: moves the object right ½ meters

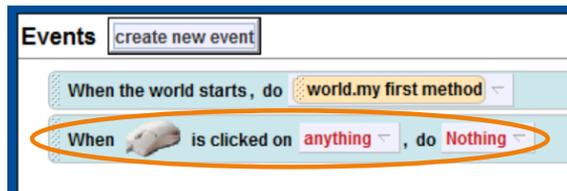
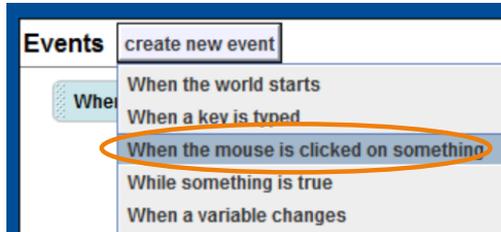
Exercise

- Create a method for monkey, head fly:



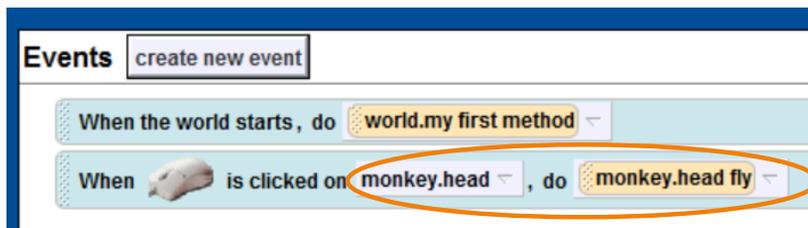
When the mouse is clicked

- Register the handler



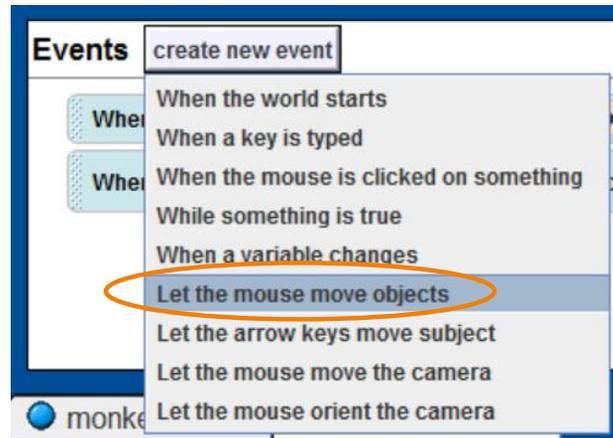
When the mouse is clicked

- Register the handler



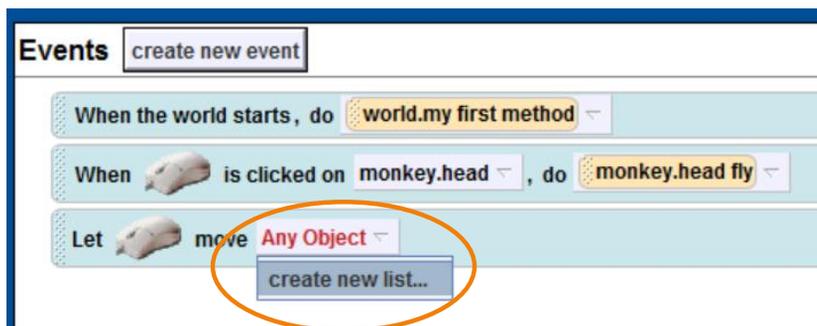
When the mouse is clicked

- Create a new event



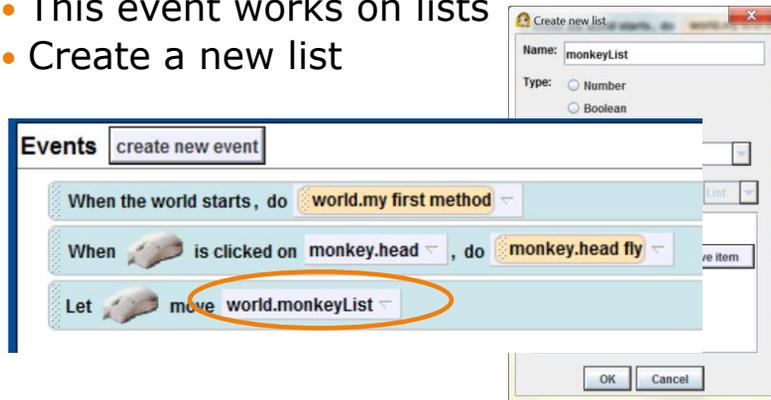
Let the mouse move an object

- This event works on lists
- Create a new list



Let the mouse move an object

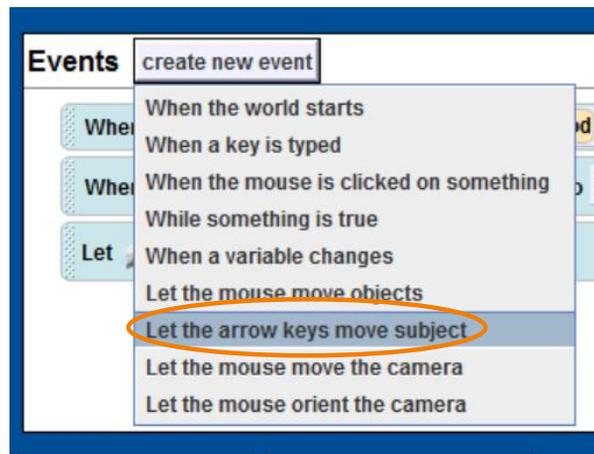
- This event works on lists
- Create a new list



- Click on the monkey to drag it

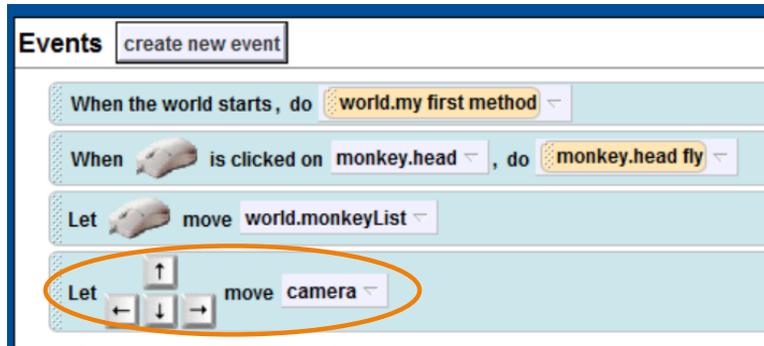
Let the mouse move an object

- Create a new event



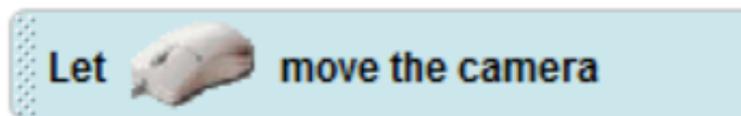
Let the arrow keys move an object

- Create a new event



Let the arrow keys move an object

- This event allows you to drag the mouse and move the camera



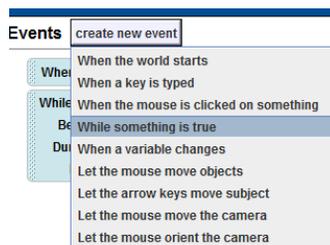
Let the mouse move the camera

- Similar to moving the camera, except it only moves the camera left or right (orientation)



Let the mouse orient the camera

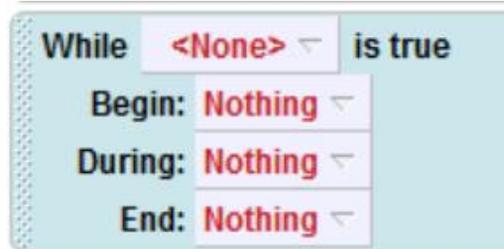
- As long as something is true, perform the handler action



- This action is performed repeatedly, until the while condition becomes false

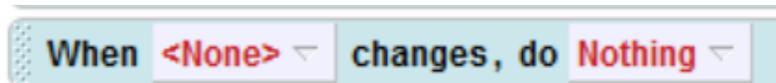
While and events

- While a condition is true do the following



- Up to three actions are performed: When the while condition becomes true, while the condition is true, and when it becomes false.

While something is true



When a variable changes

Example 17

Important variable

world's details

properties methods funct

123 highScore = 888

Define key methods

world's details

properties methods function

start edit

newHighScore edit

increaseHighScore edit

Register methods

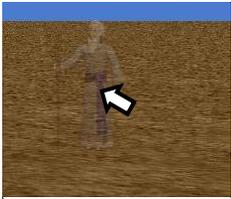
Events create new event

When the world starts, do world.start

When  is clicked on raiden, do world.increaseHighScore

When world.highScore changes, do world.newHighScore

JT: Event For Changing Variables



raiden set opacity to 1 (100%) duration = 2 seconds more...

raiden resize 2 more...

// "Flawless victory wav file from Mortal Kombat (C) NewLine Cinema

raiden play sound world.mk-flawless1 (?:?) more...

JT: Event For Changing Variables