# GridWorld Role Play
# Guidelines for Teachers

**I. "Actors" Needed**

1. **RolePlayBugRunner**
2. **Grid****
3. **ActorWorld**
4. **Bug1**
5. **Flower1 (added after Bug1 acts)**
6. **Bug2**
7. **Flower2 (added after Bug2 acts)**
8. **Flower**
9. **Rock**
10. **Critter (for second simulation)**

**II. Use GridWorld Simulation for display**

**III. Use nerfball to throw around room to show "active" object. Post-its may be posted to the ball to pass parameters.**

**IV. RC Cola! Remember for 2D Arrays, the elements are referred to in [row][column] (aka 'row-major" or left-right and top-down) notation.**

**V. Schedule**

- **a. Day 1**
    - **i. GW Overview**
    - **ii. GW Role Play is an approximation to code, not exact…no getters/setters.**
    - **iii. Follow GW Role Play rules (be the computer!)**
    - **iv. Use visible grid (Excel, whiteboard, GridWorld Simulation, etc.) and choose strategic placement of Actors to investigate "special cases."**
    - **v. Require students to say line they are on when entering a new method.**
    - **vi. Assign appropriate roles. Use understudies for large class sizes.**
    - **vii. Object construction – get through constructing all object on Day 1**
- **b. Day 2/3**
    - **i. Run through 1 ActorWorld "step" method**
- **c. Day 4**
    - **i. Add Critters to role play**
- **d. Code Walkthrough**
    - **i. Review syntax**
    - **ii. Discuss design decisions**
    - **iii. Solicit ideas of modifications to GW ("toral" world, new creatures**

**This version of the GridWorld role play has been edited with input from participants from the AP Summer Institute at Cal State San Marcos 2007, 2008, and 2009; the Silver State AP Summer Institute at Las Vegas 2007 and 2008; and Loyola High School AP Computer Science students spring 2008 and 2009.**

# You are the RolePlayBugRunner (“The Producer”)

**Run**

When the role play is set to begin

1) Construct an ActorWorld by picking the pre-selected person in the room and saying, <person name>, “**Construct** yourself as an ActorWorld”
2) Construct a Bug by picking the pre-selected person in the room and saying, <person name>, “**Construct** yourself as an Bug; your color is Red”
3) Ask your Bug to set its direction to East. (This is not in the role play script…Bug, you can just change your direction)
4) Ask your ActorWorld to **add** Bug <person name> at position (3,3)
5) Construct a Bug by picking pre-selected person in the room and saying, <person name>, “**Construct** yourself as an Bug; your color is “Blue”
6) Ask your Bug to set its direction to North. (This is not in the role play script…Bug, you can just change your direction)
7) Ask your ActorWorld to **add** Bug <person name> at position (0,0)
8) Construct a Flower by picking the pre-selected person in the room and saying, <person name>, “**Construct** yourself as an Flower”
9) Ask your ActorWorld to **add** Flower <person name> at position (5,4)
10) Construct a Rock by picking the pre-selected in the room and saying, <person name>, “**Construct** yourself as an Rock”
11) Ask your ActorWorld to **add** Rock <person name> at position (2,3)
12) Have a **commercialInterruption**
13) Ask your ActorWorld to **step**.
14) Have a **commercialInterruption**
15) Ask your ActorWorld to **step**.

## commercialInterruption

To process a **commercialInterruption**

1) Narrate "We now interrupt this role play for a commercial interruption. The sponsors hope that you will pay attention to the display during this break."
2) For each Actor (e.g. Bug, Flower, Rock, Critter – NOT ActorWorld or Grid) who is currently involved in the role play, do the following:
    a) Ask the actor if their Grid is null
        i) If the actor says, "yes", then skip to the next actor.
    b) Ask the actor for his/her location.
    c) Ask the actor for his/her color.
    d) Ask the actor for his/her direction.
    e) Update whatever display your instructor has provided (e.g. blackboard, whiteboard, poster, etc.) to reflect this actor's status in the world.
3) Narrate "No more actors. We now resume our regularly scheduled program." (Feel free to emphasize the last word…)

# ActorWorld ________________________ *(your name here)*

## Private information

**your Grid**: ___________________

## Actors List (in order of appearance)

ActorWorld: ________________

Grid: ________________

Bug: ________________

Flower: ________________

Rock: ________________

# You are an ActorWorld. (“The Director”)

## Constructing Yourself

When you are constructed you may be given:

- Your **Grid**

1) <u>If</u> you are not given this information:
   a) <u>Construct</u> a Grid by picking an unused person in the room and saying, <person name>, “**Construct** yourself as a Grid”
2) <u>Remember</u> the name of your Grid by recording it in the appropriate place on your **Private Data** sheet.
3) <u>Say</u> “Done constructing ActorWorld <*your name*>.”

## add

When asked to **add**, you will be given an Actor and a Location

1) <u>Ask</u> <*the actor you are given*> to **putSelfInGrid** <*your grid*> at <*the location you were given*>
2) <u>Say</u> “<name of the actor> inserted from perspective of ActorWorld.”

## step

When asked to **step**

1) <u>Ask</u> <your Grid> to **getOccupiedLocations** and place the occupied locations into a list (use row, column format)
2) <u>For</u> each Location in the List:
   a) <u>Ask</u> the Grid for **nameOfActorAt** each location in the list passed you from Grid. (there is not script for this, Grid will simpley return the list of names.
   b) <u>Add</u> that name to your ***<listOfNames>***
3) <u>For</u> each name in your ***<listOfNames>***
   a) <u>Ask</u> that person for the name of their Grid
   b) <u>If</u> they do not say, “blank” (meaning null)
      i) <u>Ask</u> that person to **act**
4) <u>Say</u> “Done stepping.”

**listOfNames**

# Grid ___________________________ *(your name here)*

**Private information (aka “who is where”)**

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | | | | | | | | | | |
| 1 | | | | | | | | | | |
| 2 | | | | | | | | | | |
| 3 | | | | | | | | | | |
| 4 | | | | | | | | | | |
| 5 | | | | | | | | | | |
| 6 | | | | | | | | | | |
| 7 | | | | | | | | | | |
| 8 | | | | | | | | | | |
| 9 | | | | | | | | | | |

# You are a Grid. (“The Stage”)

### Constructing Yourself

1) Verify that the table you have (on paper) is currently empty.
2) Say “Done constructing Grid <*your name*>.”

### isAValidLocation

When asked **isAValidLocation**, you will be given a Location

:

1) If the value of row in that location is between 0 and 9 inclusive AND the value of the column in that location is also between 0 and 9 inclusive.
    a) Say “Yes”

    Otherwise
    a) Say “No”

### get

When asked to **get**, you will be given a Location

1) If there is nothing in that cell in your current, private table
    a) Say “null”

    Otherwise:
    a) Say “<*The name of the Object at that Location in your private table*>.”

### put

When asked to **put**, you will be given an Actor and Location

1) Write the name of the Actor you were given into the specified location in your current, private table. Erase any name that was there before this operation.
2) Say, “<*name of actor*> has been placed in location <*given location*>.”

# You are a Grid.

---

**remove**

When asked to **remove**, you will be given a Location

1) If there is no Actor at the given Location, then **Say** “There is no Actor to remove, done removing”.
2) Erase the name of the Actor you were given into the specified location in your current, private table, remembering the name erased for just a moment.
3) Say, “<*name of actor erased*> has been removed from location <*given location*>.”

---

**getOccupiedLocations**

When asked to **getOccupiedLocations**

1) On your Post-it note, write down all of the locations (coordinate pairs, not names) that are not null according to your current, private table.
2) Give that Post-it note to the person who made the request, *while* Saying, “Here is the list of occupied locations.”

# You are a Grid.

## getNeighbors

When asked to **getNeighbors**, you will be given a Location

1) Narrate, “I will begin by getting a list of *occupied* adjacent locations by calling my helper method **getOccupiedAdjacentLocations**.”
2) Do **getOccupiedAdjacentLocations** – this method is located below.
3) Take a blank piece of paper (call it the Actors List sheet.)
4) Narrate, “I will now get the name of the actor associated with each location.”
5) On your new Actors List sheet, write down the *name* of each actor that is associated with a location on the sheet from Step 2 as indicated by your private table.
6) Crumple up the sheet from Step 2.
7) Narrate, “I have now constructed the list of neighbors.” and Hold the sheet from Step 5 up for the audience to see.
8) If the request for this list came from someone else, give that person the list.

## getOccupiedAdjacentLocations

When asked to **getOccupiedAdjacentLocations**, you will be given a Location

1) Narrate, “I will begin by getting a list of *valid* adjacent locations by calling my helper method **getValidAdjacentLocations**”
2) Do **getValidAdjacentLocations** (below).
3) Take a blank Location List sheet.
4) Narrate, “I will now copy over only those locations that are occupied.”
5) On your new Location List sheet, write down all of the locations (coordinate pairs, not names) from the valid locations sheet that are currently occupied according to your private table.
6) Crumple up the sheet from Step 2 (the first sheet that you made)
7) Narrate, “I have now constructed the list of occupied adjacent locations.” and Hold the sheet from Step 5 up for the audience to see.
8) If the request for this list came from someone else, give that person the list. I am done!

# You are a Grid.

### getValidAdjacentLocations

When asked to **getValidAdjacentLocations**, you will be given a Location

1) Take a blank Location List sheet.
2) On that sheet, write down all of the locations (coordinate pairs, not names) from your table that are adjacent to the given location. DO NOT include any locations that are outside your table due to coordinate values being too low or too high.
3) Narrate, "I have now constructed the list of valid adjacent locations." and Hold it up for the audience to see.
4) If the request for this list came from someone else, give that person the list.

### getEmptyAdjacentLocations

When asked to **getEmptyAdjacentLocations**, you will be given a Location

1) Narrate, "I will begin by getting a list of *valid* adjacent locations by calling my helper method **getValidAdjacentLocations**()"
2) Do **getValidAdjacentLocations**() – this is located above
3) Take a blank Location List sheet.
4) Narrate, "I will now copy over only those locations that are empty."
5) On your new Location List sheet, write down all of the locations (coordinate pairs, not names) from the valid locations sheet that are currently empty according to your private table.
6) Crumple up the sheet from Step 2.
7) Narrate, "I have now constructed the list of empty adjacent locations." and Hold the sheet from Step 5 up for the audience to see.
8) If the request for this list came from someone else, give that person the list.

**Location List sheet**

(______,_______)     (______,_______)
(______,_______)     (______,_______)
(______,_______)     (______,_______)
(______,_______)     (______,_______)
(______,_______)     (______,_______)

# Bug ______________________ *(your name here)*

## Private information

yourGrid: ________________

yourDirection: ________________

yourColor: ________________

yourLocation: ( _______ , _______ )

# You are a Bug. (“The Actor”)

## Constructing Yourself

When you are **constructed** you will be given:

- Your **Color**

1) Remember your color by recording it in the appropriate places on your **Private Data** sheet. (Note: if you were not given your color, then assume that your color is “Red” and your direction is “North”.) Leave other data blank for now.
2) Say “Done constructing Bug <*your name*>.”

## Are you a _______?

When asked if you are a **Bug** or an **Actor**:

1) Say “true”.

When asked if you are anything else (i.e., anything other than a **Bug** or an **Actor**):

1) Say “false”.

## act

When asked to **act**:

1) Narrate “Hmm… I must determine if I can move.”
2) Ask yourself if you **canMove (this script is on the next page)**.
3) If the answer was that you can move
   a) then **move (this script is on the next page)**
   Otherwise
   a) **turn (skip two pages to find this script)**
4) Say “<*Your name*> is done acting.”

# You are a Bug.

---

**canMove**

When asked if you **canMove**

1) Say "My location is <*yourlocation*> (row, column)
2) Say "My direction is <*your direction*>
3) Say "Therefore, my next location (row, column) would be <*the result of moving forward*>"
4) Name this new Location <*nextLocation***>**.
5) Ask your grid if <*nextLocation*> **isAValidLocation**
6) If the grid says, "No", then say, "I cannot move" (and return to your **act** script on the previous page.)
7) Ask your grid to **get** the name of the object (if not null) at location ***<nextLocation>.***
8) If the grid says, "Null", then Say, "I can move" (and return to your act script)
9) Otherwise Ask that person, "Are you a Flower?"
10) If that person says, "Yes", then Say, "I can move" (and return to your **act** script on the previous page)
11) If that person says, "No", then Say, "I cannot move" (and return to your **act** script on the previous page)

**nextLocation**
**(_____,_____)**

**Return to act method script, line 3**

---

**move**

When asked to **move**:

**currentLocation** **(_____,_____)**    **nextLocation** **(_____,_____)**

1) If your grid is null
   - a) Say "Done Moving." (…and return to directions for **act**)

   Otherwise:
   - a) Make a copy of <*yourLocation*> and name it <*currentLocation*>Write this in the box above.
   - b) Determine the Location in front of you as described below:
     - i) Say "My location is <*your location*>
     - ii) Say "My direction is <*your direction*>
     - iii) Say "Therefore, my next location (row, column) would be <*the result of moving forward*>"
     - iv) Name this next Location <*nextLocation***>**.
   - c) Ask your Grid if ***<nextLocation>*** **isAValidLocation**.
   - d) If the Grid replies positively
     - i) **moveTo** the location <*nextLocation***>** (script on next page)

     Otherwise
     - i) **removeYourself** from the grid.
   - e) Pick a person not currently involved in the role play and tell them to construct themselves as a Flower with color <*yourColor*>.
   - f) Tell that person to **putSelfInGrid** into **<your grid>** at location <*currentLocation***>**.
   - g) Narrate "I am done moving." (return to **act** script above)

**Why are we asking the Grid "isAValidLocation()" again? Didn't we just do that?**

**How does ActorWorld know that a new Flower is now in the Grid?**

# You are a Bug.

### moveTo

**nextLocation**
**(_____,_____)**

When asked to **move**, you will be given a **<*nextLocation*>**. (If not, complain.)

1) Tell your grid to **remove** the object at **<*currentLocation*>**. That object may stand up and step out of the grid to simulate being removed from the grid.
2) Tell the object at **<*nextLocation*>** (if any) to **removeYourself** (see script below). This object may also stand up and step out of the grid to simulate being removed from the grid.
3) Set **<*your location*>** to **<*nextLocation*>**, remembering to write it on your private data sheet. I am setting my location to <*nextLocation*>.
4) Tell your grid to **put** <*your name*> at <*your location*>

**Return to move method script, line 1di**

### turn

When asked to **turn**:

1) Narrate "My current direction is <*your current direction*>."
2) Narrate "The direction 45 degrees clockwise from that is <*the appropriate direction*>."
3) Set your direction to that direction, remembering to write it on your private data sheet.
4) Say, "I am done turning."

**Return to the method that called turn()**

### putSelfInGrid

When asked to **putSelfInGrid,** you will be a given a grid and location. Write these on your private data sheet.

1) Narrate "I need to be placed into the grid"
2) Ask <*your grid*> to **"put"** <*your name*> at location <*the location you were given*>
3) Say "Done putting myself into the grid."

### removeYourself

When asked to **removeYourself**

1) Narrate "I need to get out of the grid, but I can't do this alone."
2) Ask your grid to **remove** <*your name*>
3) Set your grid to null
4) Set your location to null
5) Say "Done removing myself from the grid."

# Rock ________________________ *(your name here)*

## Private information

**your Grid**: ___________________

**your Color**: ___________________

**your Location**: ( ________ , ________ )

# You are a Rock.

---

**Constructing Yourself**

When you are **constructed** you might be given your color. If not given your color, assume that it is black.

- Your **Color**

1) Remember your color by recording it in the appropriate places on your **Private Data** sheet. (Note: if you were not given your color, then assume that your color is "Black" and direction "North"). Leave other data blank for now.
2) Say "Done constructing Rock <*your name*>."

---

**Are you a _______?**

When asked if you are a **Rock** or an **Actor**:

1) Say "true".

When asked if you are anything else (i.e., anything other than a **Rock** or an **Actor**):

1) Say "false".

---

**act**

When asked to **act**

1) Say "<*Your name*> is done acting."

# You are a Rock.

### putSelfInGrid

When asked to **putSelfInGrid,** you will be a given a grid and location. Write these on your private data sheet.

1) Narrate "I need to be placed into the grid"
2) Ask your grid to "**put"** <*your name*> at location <*the location you were given*>
3) Say "Done putting myself into the grid."

### removeYourself

When asked to **removeYourself**

1) Narrate "I need to get out of the grid, but I can't do this alone."
2) Ask your grid to **remove** <*your name*>
3) Set your grid to null
4) Set your location to null
5) Say "Done removing myself from the grid."

# Flower ________________________ *(your name here)*

## Private information

**your Grid**: ____________________

**your Color**: ____________________

**your Location**: ( ________ , ________ )

# You are a Flower.

## Constructing Yourself

When you are **constructed** you will be given:

- Your **Color**

1) Remember your color by recording it in the appropriate places on your **Private Data** sheet. (Note: if you were not given your color, then assume that your color is “Pink” and direction “North”). Leave other data blank for now.
2) Say “Done constructing Flower <*your name*>.”

## Are you a _______?

When asked if you are a **Flower** or an **Actor**:

1) Say “true”.

When asked if you are anything else (i.e., anything other than a **Flower** or an **Actor**):

1) Say “false”.

## act

When asked to **act**:

1) Narrate “Hmm… I must darken myself.”
2) Using the table below, set your color to the next darkest color. If you are already at “Black” just stay there.
3) Narrate “My new color is <*your color*>.
4) Say “<*Your name*> is done acting.”

Darkening table: Assumes you start at “red”

| |
|---|
| Pink |
| Red |
| Deep red |
| Slightly dark red |
| Pretty dark red |
| Dark red |
| Very dark red |
| Reddish black |
| Black with a hint of red |
| Essentially black |
| Black |

# You are a Flower.

---

**putSelfInGrid**

When asked to **putSelfInGrid,** you will be a given a grid and location. Write these on your private data sheet.

1) Narrate "I need to be placed into the grid"
2) Ask your grid to "**put"** <*your name*> at location <*the location you were given*>
3) Say "Done putting myself into the grid."

---

**removeYourself**

When asked to **removeYourself**

1) Narrate "I need to get out of the grid, but I can't do this alone."
2) Ask your grid to **remove** <*your name*> at location <*your location*>.
3) Set your grid to null
4) Set your location to null
5) Say "Done removing myself from the grid."

# You are the RolePlayCritterRunner

**Run**

When you the role play is set to begin

1) Construct an ActorWorld by picking a pre-selected person in the room and saying, <person name>, “Construct yourself as an ActorWorld”
2) Construct a Critter by picking a pre-selected person in the room and saying, <person name>, “Construct yourself as an Critter”
3) Ask your ActorWorld to **add** the Critter <person name> at position (0,0)
4) Construct a Rock by picking a pre-selected person in the room and saying, <person name>, “Construct yourself as an Rock”
5) Ask your ActorWorld to **add** your Rock <person name> at position (1,0)
6) Construct a Bug by picking a pre-selected person in the room and saying, <person name>, “Construct yourself as an Bug; your color is Blue”
7) Ask your ActorWorld to **add** your Bug <person name> at position (0,1)
8) Construct a Bug by picking a pre-selected person in the room and saying, <person name>, “Construct yourself as an Bug; your color is Yellow”
9) Ask your ActorWorld to **add** your Bug <person name> at position (1,2)
10) Construct a Critter by picking a pre-selected person in the room and saying, <person name>, “Construct yourself as an Critter”
11) Ask your ActorWorld to **add** your Critter <person name> at position (2,2)
12) Construct a Bug by picking a pre-selected person in the room and saying, <person name>, “Construct yourself as a Bug”
13) Ask your ActorWorld to **add** your Bug <person name> at position (3,2)
14) Have a **commercialInterruption**
15) Ask your ActorWorld to **step**.
16) Have a **commercialInterruption**
17) Ask your ActorWorld to step.

# You are the RolePlayCritterRunner

**commercialInterruption**

To process a **commercialInterruption**

1) Narrate "We now interrupt this role play for a commercial interruption. The sponsors hope that you will pay attention to the display during this break."
2) For each Actor (e.g. Bug, Flower, Rock, Critter – NOT ActorWorld or Grid) who is currently involved in the role play, do the following:
   a) Ask the actor if their Grid is null
      i) If the actor says, "yes", then skip to the next actor.
   b) Ask the actor for his/her location.
   c) Ask the actor for his/her color.
   d) Ask the actor for his/her direction.
   e) Update whatever display your instructor has provided (e.g. blackboard, whiteboard, poster, etc.) to reflect this actor's status in the world.
3) Narrate "No more actors. We now resume our regularly scheduled program." (Feel free to emphasize the last word…)

# Critter ________________________ *(your name here)*

## Private information

**your Grid**: ____________________

**your Direction**: ____________________

**your Color**: ____________________

**your Location**: ( ________ , ________ )

# You are a Critter.

---

### Constructing Yourself

When **constructed**, assume that your color is "Blue" and your direction is "North."

1) Remember your color and direction by recording it in the appropriate places on your **Private Data** sheet.
2) Say "Done constructing Critter <your name>."

---

### Are you a _______?

When asked if you are a **Critter** or an **Actor**:

1) Say "true".

When asked if you are anything else (i.e., anything other than a **Critter** or an **Actor**):

1) Say "false".
2)

---

### putSelfInGrid

When asked to **putSelfInGrid,** you will be a given a grid and location. Write these on your private data sheet.

1) Narrate "I need to be placed into the grid"
2) Ask your grid to "**put"** <*your name*> at location <*the location you were given*>
3) Say "Done putting myself into the grid."

---

### removeYourself

When asked to **removeYourself**

1) Narrate "I need to get out of the grid, but I can't do this alone."
2) Ask your grid to **remove** <*your name*>
3) Set your grid to null
4) Set your location to null
5) Say "Done removing myself from the grid."

# You are a Critter.

**act**

When asked to **act**:

1) Narrate "Ahh, to act… First I will begin by getting a list of the objects with whom I might interact."
2) Narrate "Ahh, I must ask myself to **getActors** by calling my helper method **getActors**." (Jump to that script on the next page)
3) Narrate "Behold my list of actors!! Now to continue action. Since I have the list of actors, I must now process it."
4) Narrate "Ahh, I must ask myself to **processActors** by calling my helper method **processActors**. (Jump to that script on the next page)
5) Narrate "Now that I am done processing everyone, I can think about moving. I will start by getting list of possible locations.
6) Narrate "Ahh, I must ask myself to **getMoveLocations** by calling my helper method **getMoveLocations**. (Jump to that script on the next page)
7) Narrate "Ahh, the list of locations…now I must choose one…"
8) Narrate "Ahh, I must ask myself to **selectMoveLocation** by calling my helper method **selectMoveLocation**. (Jump to that script on the next page)
9) Narrate "I have chosen <*the chosen location*> and will now make my move"
10) Narrate "Ahh, I must ask myself to **makeMove** to <*the chosen location*> by calling my helper method **makeMove**. (Jump to that script on the next page)
11) Narrate "At long, last, I am done acting."

# You are a Critter.

### getActors

When asked to **getActors**

1) Narrate “Hmmm… With whom do I act? With my neighbors, of course. I’d better ask the grid who they are.”
2) Ask <*your grid*> to **getNeighbors** of <*your location*>
3) Wave the list given to you by the grid and narrate “Behold my list of actors!”

[Return to Step 3 of **act**.]

### processActors

When asked to **processActors**, you should already have a list of actors…

1) Narrate “Time to eat… Who is near by that is edible?”
2) For each actor in your list
    a) Ask <*that actor*> if it is a Rock.
        i) If it says, “Yes”, dramatically cross it off the list and move to the next actor in the list.
    b) Ask <*that actor*> if it is a Critter.
        i) If it says, “Yes”, dramatically cross it off the list and move to the next actor in the list.
    c) Ask <*that actor*> to **removeYourself**
    d) Dramatically cross <*that actor*> off the list and move to the next actor in the list.
3) Narrate “That’s the whole list!”

[Return to Step 5 of **act**.]

# You are a Critter.

### getMoveLocations

When asked to **getMoveLocations**

1) Narrate "Hmmm… Where can I move? To any adjacent empty space. I'd better ask the grid which ones those are."
2) Ask <*your grid*> to **getEmptyAdjacentLocations** of <*your location*>
3) Wave the list given to you by the grid and narrate "Behold my list of possible destinations!"

[Return to Step 7 of **act**.]

### selectMoveLocation

When asked to **selectMoveLocation**, you should already have a list of locations…

1) Narrate "Time to move…There are <*number of items in your list*> possibilities."
2) Ask <*The Random Number Generator*> for a number up to <*number of items in your list*>
3) Narrate "Let's see. The <*number you were given*>th item in my list is <*that location*>. I'll move there!"

[Return to Step 9 of **act**.]

Why is there a separate makeMove method? Why not just call moveTo?

### makeMove

When asked to **makeMove**, you will be given a location.

1) Narrate "Time to move…my choice is to move to <*that location*>."
2) Narrate "Actually, I do this the same way any actor does…"
3) Ask yourself to **moveTo** <*that location*>. (see script below)

[Return to Step 11 of **act**.]

---

### moveTo

When asked to **moveTo**, you will be given a **newLocation**. (If not, complain.)

1) Tell <*your grid*> to **remove** the object at <*your current location*>.
2) Tell the object at <***nextLocation***> (if any) to **removeYourself** (see script below). This object may also stand up and step out of the grid to simulate being removed from the grid.
3) Set <*your location*> to <newLocation**>**, remembering to write it on your private data sheet.
4) Tell your grid to **put** <*your name*> at <*your location*>
g) Say "I have completed my move and will return to my **makeMove** script.