



Being in the Game: Implementing First Person

Adding Challenge to a game like Journey or PacMan by switching viewpoints from bird's eye to first person so that the player only sees what the Traveler or PacMan sees rather than the entire game world.

Created by: Catharine Brand and Susan Miller, University of Colorado

This curriculum has been designed as part of the Scalable Games Design project.
It uses material from Fred Gluck's video tutorials and Susan Miller's SGD curricula.

This material is based upon work supported by the National Science Foundation under Grant No. DRL-1312129 and CNS-1138526. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Vocabulary/Definitions

Agent Attributea value assigned to an agent (such as scent)

Bird's Eye View..... looking down on a World as if you were flying over it

First Person View.... seeing a world from the point of view of a single agent

Local Variablea variable (attribute) belonging to a specific agent

Mod or Modulothis mathematical operation returns the remainder when one number is divided by another

STUDENT Handout A:

Create a world for First Person Navigation

Creating a simple world for exploring first person movement

You will create a world to learn how First Person Navigation works in AgentCubes Online. To do this, you will create a new world that is occupied by a ladybug. We will add the first person rules to a ladybug because it has a head end, which makes it much easier to see that it turns and moves correctly in response to each arrow key. Testing our rules is easier if we work in a simplified world where we can instantly see which way our agent is facing. You can choose to add these rules to an existing game that you want to change into First Person, or create a brand new game as an example.

- Make a new level called First Person in the game in which you would like to use the first person perspective.
- Select PacMan or the Traveler or Frogger or any main character who is keyboard-controlled with the big arrow tool, click on the +Shape button to make a 2nd shape and name it ladybug.
 - Choose Inflatable Icon, the Animals category and then Akako, the ladybug.
- **Or** create a new agent named Ladybug and choose the Akako shape.

Create the following 6 agents:

1. An agent with a clear front and back (we strongly encourage you to use Ladybug and choose the Akako shape)
2. a tile named “Ground”
3. Select a cube for the north wall by choosing the Letters-Bit font category and then picking n.png.
4. Select a cube for the west wall by choosing the Letters-Bit font category and then picking w.png.
5. Select a cube for the south wall by choosing the Letters-Bit font category and then picking s.png.
6. Select a cube for the east wall by choosing the Letters-Bit font category and then picking e.png.

Create your first person world:

1. Cover it with a layer of Ground tiles.
2. Put the Ladybug in the center of the World with her head facing the top of the World.
3. Draw a line of north blocks across the top of the World.
4. Draw a line of west blocks down the left side of the World.
5. Draw a line of south blocks across the bottom of the World.
6. Draw a line of east blocks up the right side of the World.

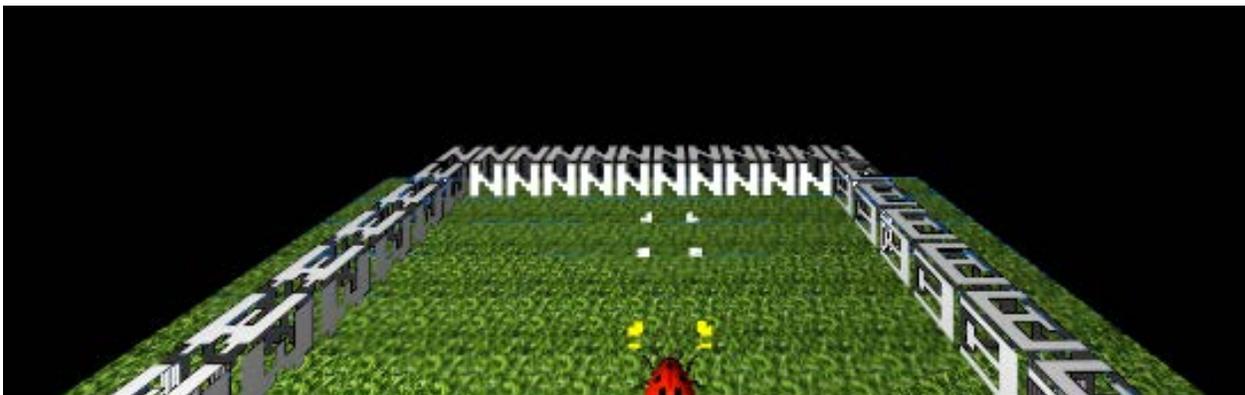
Your First Person World should look like this:



Exploring first person movement with bird's eye perspective rules

- Set up the Ladybug with the basic key-controlled movement rules so the 4 arrow key rules move it up, down, left, and right.
- Select the Ladybug with the big arrow key, then click on the camera to switch to first person. Use the Rotate , pan  and zoom  tools to adjust the camera so that it is looking over the top of the agent's head, as if the agent were wearing a head-mounted GoPro camera. This lets the viewer see the world from the agent's point of view.

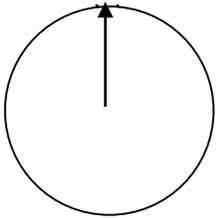
Here is what the first person world should look like when the Ladybug is facing up and the camera has been set to look over her head:



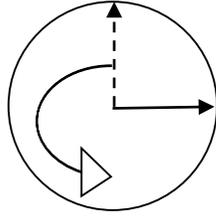
Run the game. In what ways is the movement realistic. In what ways is it not?

Student Handout B: Connecting Ladybug Movement to Modulo Math

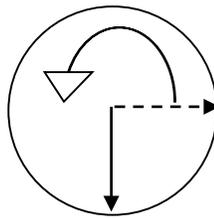
Let's say the Ladybug is facing up (0°), and the player clicks the \rightarrow button (270 degrees) three times, which way is the ladybug facing?



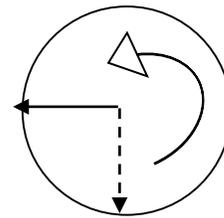
Starts here
Starts at 0°



Rotates 270°
 $+ 270^\circ$



Rotates 270°
 $+ 270^\circ$



Rotates 270°
 $+ 270^\circ = 810^\circ$

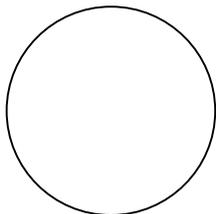
According to the math, the ladybug travels to 810° , but if we look at the picture, we can see the ladybug is facing 90° .

Mathematically, we can show that also, using the mod function

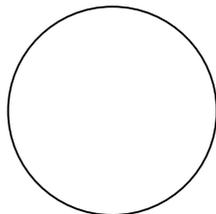
$$810 \div 360 = 2, \text{ Remainder } 90$$

90 is the ending direction of the lady bug.

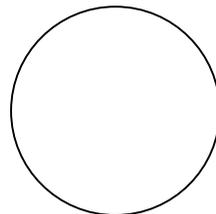
If my ladybug is facing down (180°), and the player clicks the \leftarrow button (90 degrees) three times, which way is the ladybug facing?



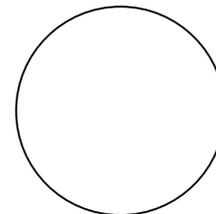
Starts here



Rotates _____ $^\circ$



Rotates _____ $^\circ$



Rotates _____ $^\circ$

Starts at 180°

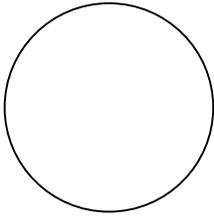
$+ \text{_____}^\circ$

$+ \text{_____}^\circ$

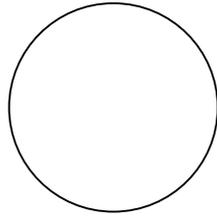
$+ \text{_____}^\circ = \text{_____}^\circ$

_____ $\div 360 = \text{_____}$, Remainder _____ **Does your remainder match your picture?**

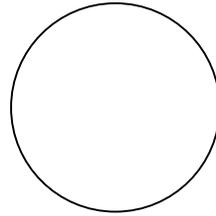
If my ladybug is facing up (0°), and the player clicks the \downarrow button (180 degrees) two times, which way is the ladybug facing?



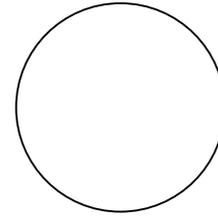
Starts here



Rotates _____ $^\circ$



Rotates _____ $^\circ$



Rotates _____ $^\circ$

Starts at 0°

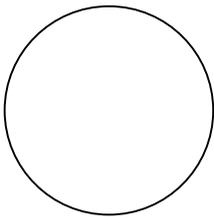
+ _____ $^\circ$

+ _____ $^\circ$

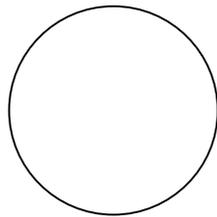
+ _____ $^\circ$ = _____ $^\circ$

_____ \div 360 = _____, **Remainder** _____ **Does your remainder match your picture?**

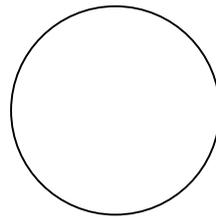
If my ladybug is facing left (90°), and the player clicks the \rightarrow button (270 degrees) three times, which way is the ladybug facing?



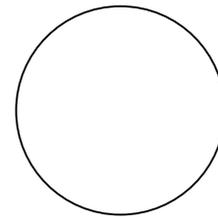
Starts here



Rotates _____ $^\circ$



Rotates _____ $^\circ$



Rotates _____ $^\circ$

Starts at 90°

+ _____ $^\circ$

+ _____ $^\circ$

+ _____ $^\circ$ = _____ $^\circ$

_____ \div 360 = _____, **Remainder** _____ **Does your remainder match your picture?**

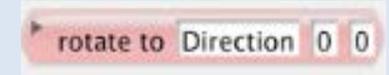
We must use the mod 360 operation because the player may type one arrow key lots of times, making the Ladybug spin in circles and adding the value of the turn to Direction over and over. AgentCubes cannot rotate an agent to a direction that is 360 or larger!

To use the mod operation, we would write:

Direction%360 if we want the remainder when we divide the direction by 360.

Talk with a partner. How does this change your ideas about creating rules for first person navigation?

Using Modulo 360 keeps the Direction Variable in the set of values {0,..., 359} so that



always turns the agent to a real position.

Student Handout C: Create the Rules for the First Person Navigation Method Using Pre-Existing Code

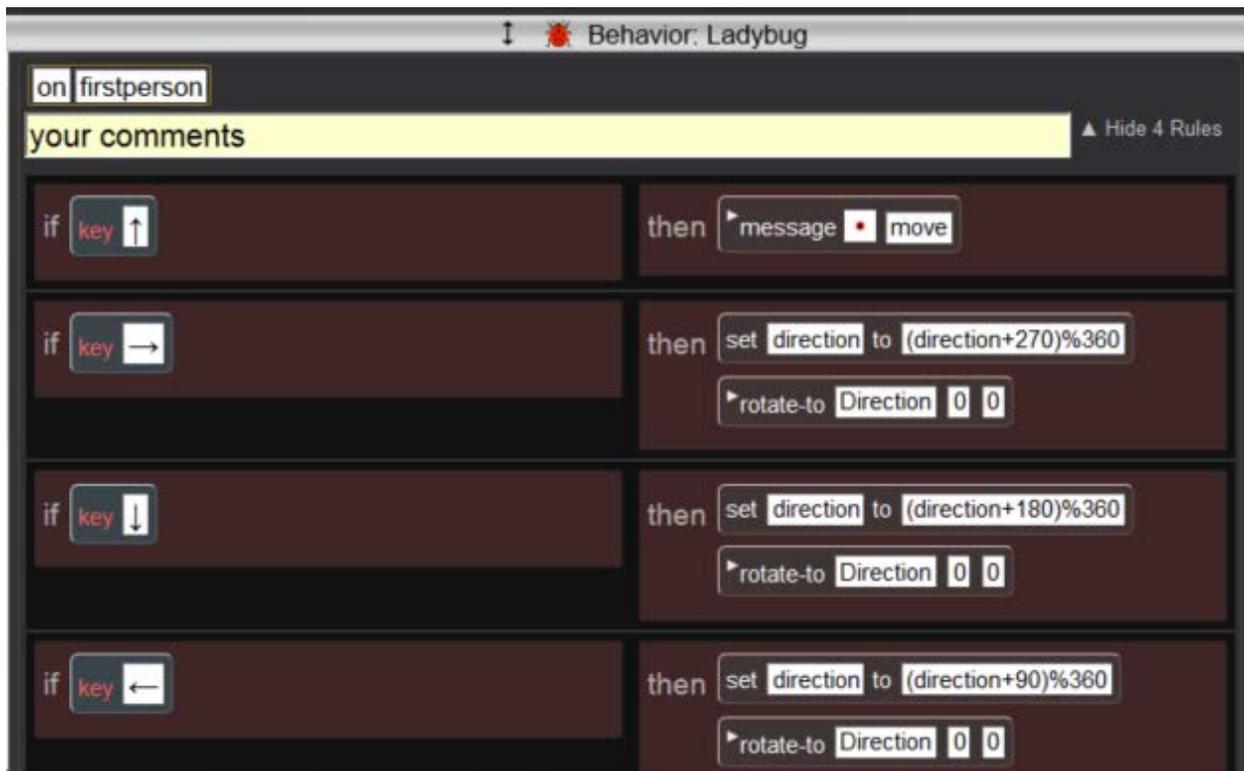
Sometimes we want to know how to do something but don't know where to start. Even with a basic understanding of what we want to accomplish (ideally, the left, right and down arrows would turn the agent, while the up arrow would move the agent forward in whatever direction it faces) and knowing something about how one might go about doing this (using modulo) may not be enough to get us started. When that happens, we might use someone else's code. Yet it's never a good idea to just insert someone else's code into your project without trying to understand it. In this portion of the lesson, we will give you the code, and then ask you to make sense of the code.

The first thing I noticed when I looked at another game that worked the way I wanted mine to work was this code. It was the first line of the 'While Running' method of the Ladybug.



Add this code to your project. What do you think this line of code will do? Why do you think it's the first line of code?

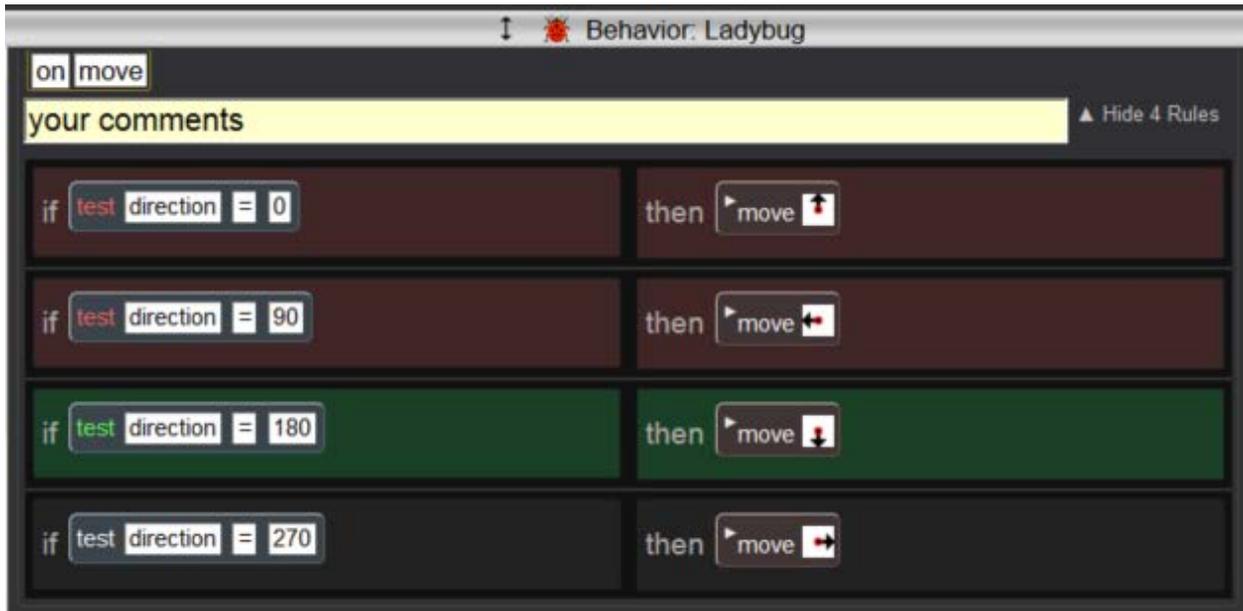
I then looked for the method that was called by that line of code. Here's what I saw.



Add this code as a new method for your ladybug. Why is the ↑ key rule different than the others?

The symbol “%” is the symbol for Modulo. What math is being done here?

This is the method ‘move’ that was called when the player uses the up arrow.



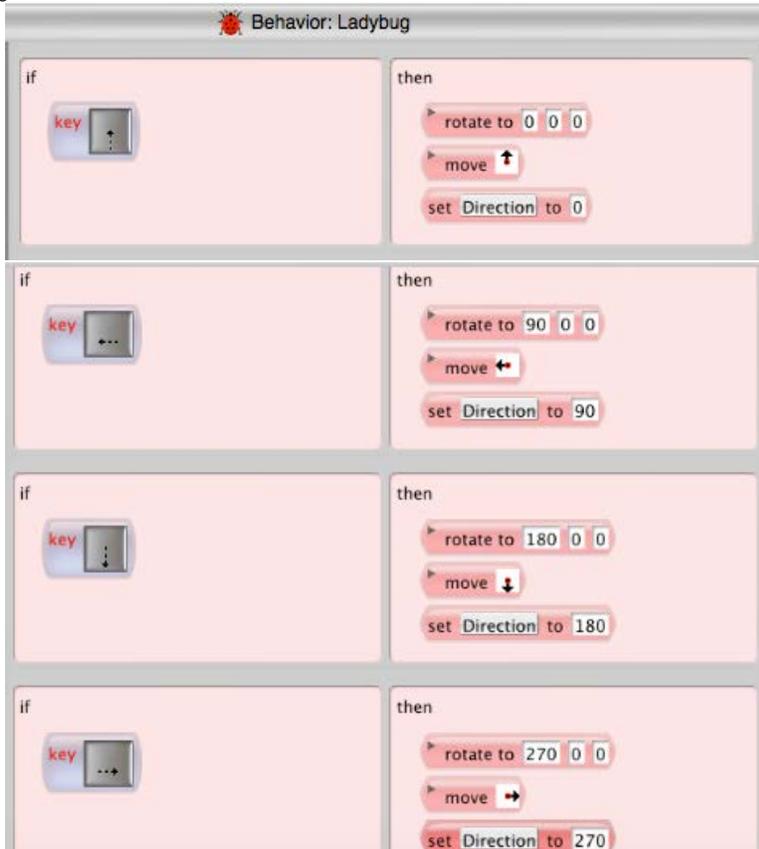
Add this code to your ladybug as a new method. Describe what’s happening here.

There were some other changes to their code. I saw this added to the Ladybug.



Add this code to your ladybug as a new method. Why is this code needed when I create a new ladybug agent?

Finally, I saw that the original movement code was also changed. Here is the code from the Ladybug's 'while running' method. This code was AFTER the first rule



Change your arrow keys to match this code. Remember, this is in the 'While Running' method. Why is this needed? Hint: It would only be active code if the FIRST PERSON NAVIGATION rule is FALSE.

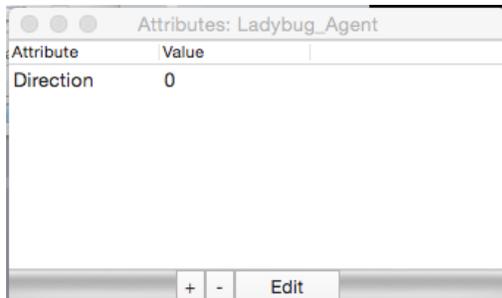
Test out your game. Does it work as expected? How can you use this to make PacMan or Journey work the same way?

Student Handout: TROUBLESHOOTING GUIDE

Test Your First Person Move Rules

Use the big arrow tool to double click on the Ladybug agent.

The Agent Attribute window should appear:



Remember what the direction numbers mean:

Turn the game on and move the Ladybug around in bird's eye view and first person **one move at a time**.

Check that the value of Direction matches the way the Ladybug is facing.

Directions for rotate to and rotate by actions:

0 degrees is **North or up**
 90 degrees is **West or left**
 180 degrees is **South or down**
 270 degrees is **East or right**

If you find a mistake, go back and check the rules for misplaced arrows or incorrect direction values. Compare your rules to the pictures above.