

Ice Arrows 2.0 Tutorial

Prerequisite: Ice Arrows 1.0 must be completed and tested! The Traveler should shoot ice arrows upwards that freeze unfrozen Chasers and unfreeze frozen Chasers.

Design Challenge: The Traveler turns to face whichever direction s/he is moving: up, down, left or right. Make the Traveler fire ice arrows at the Chasers in whichever direction the Traveler is facing. Ice arrows will move in the direction that they are fired: up, down, left or right.

Gamelet Design Activity: read the challenge and identify the new depictions which must be added to each agent.

Create New Depictions

Ice Arrow: Add new depictions so that the ice arrow moves point-first in all 4 directions

- Select the ice arrow agent and click on the New Depiction button at the bottom of the gallery window and draw a new depiction. Or go to the Gallery menu (to the right of the word AgentSheets) and find the Duplicate Depiction option. Pick the 3 different rotations of your upwards-facing ice arrow off the list of choices.
- Make sure to make them large enough and different enough that you can identify them from a small picture.
- Make an up-facing arrow, a down-facing arrow, a left-facing arrow and a right-facing arrow.
- The ice arrow's depiction stores the ice arrow's **state**: which direction it moves.

Traveler: Add new depictions to the Traveler so s/he faces in the direction s/he is moving:

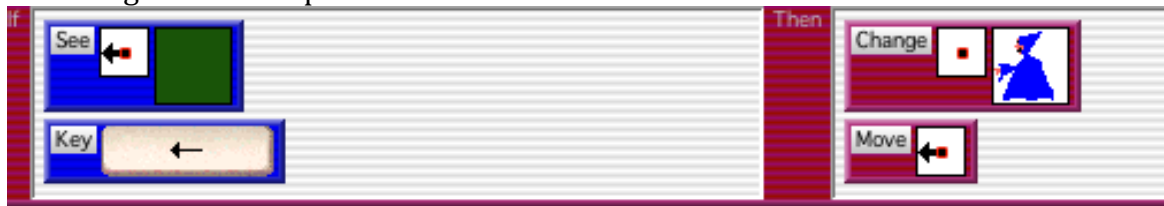
- The direction that the Traveler faces determines which ice arrow will be generated.
- The Traveler's depiction stores the Traveler's **state**: which way s/he is facing.
- Go to the Gallery menu (to the right of the word AgentSheets) and find the Duplicate Depiction option. Find the 3 different rotations of your Traveler on the list of choices. You should have an up-facing Traveler, a down-facing Traveler, a left-facing Traveler and a right-facing Traveler when you are done.

Create New Rules for the Traveler:

Edit the Traveler's rules so the depiction changes each time the Traveler moves a different direction:

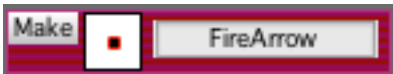
- Add an action to each of the Traveler's move rules so that the Traveler's depiction changes to match the direction of movement. For example, typing the left arrow key changes the Traveler's depiction to a left-facing depiction and the Traveler moves one square left. The Change action should be read as "Change me (the dot) to the

left-facing Traveler depiction”.



Change the Traveler's rule to fire the ice arrow:

- Use a method to decide which direction the arrow should be fired because now there are 4 possibilities to choose from.
- Remove the action from the rule with the key = spacebar condition and replace it

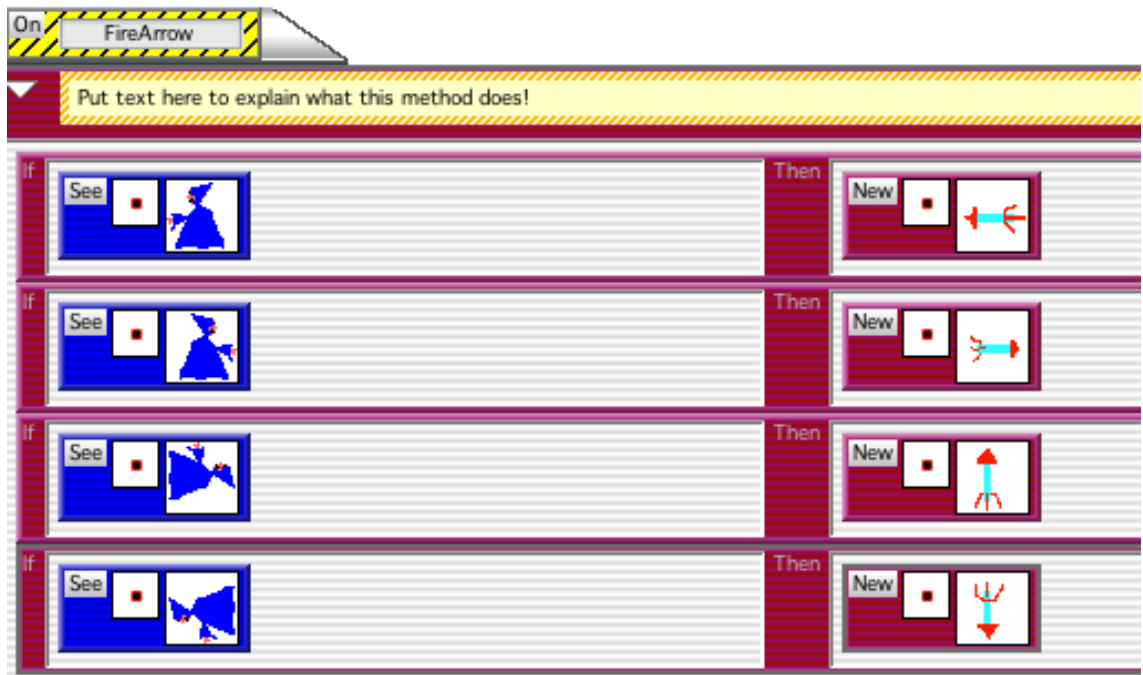
with  which should be read as “make **me** do the method named FireArrow”. The dot means “me”. The rule looks as follows:



Create the FireArrow method in the Traveler's rule window and add 4 rules to it:

- Click on the New Method button at the bottom of the Traveler's rule window.
- Edit the method name so it exactly matches the name in the Make action!
- Make the first rule by adding a condition that checks which way the Traveler is facing and then adding an action to generate a new ice arrow facing the same way.
- You may either generate the new ice arrow right on top of the Traveler by using the dot or make it appear next to the Traveler by editing the dot to be an arrow facing the same way as the Traveler.

- Duplicate this rule 3 times and edit the duplicated rules so that they generate new ice arrows in the other 3 directions.



Create New Rules for the Ice Arrow:

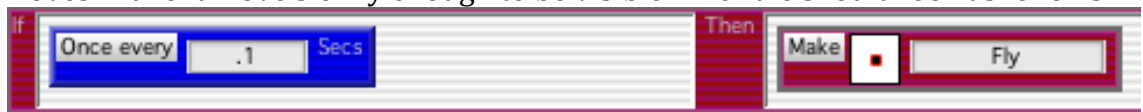
Change the Ice Arrow's movement rule so that it calls a method with rules that check which way the arrow should move:

- Change the original up-arrow move rule in the ice arrow while-running method by



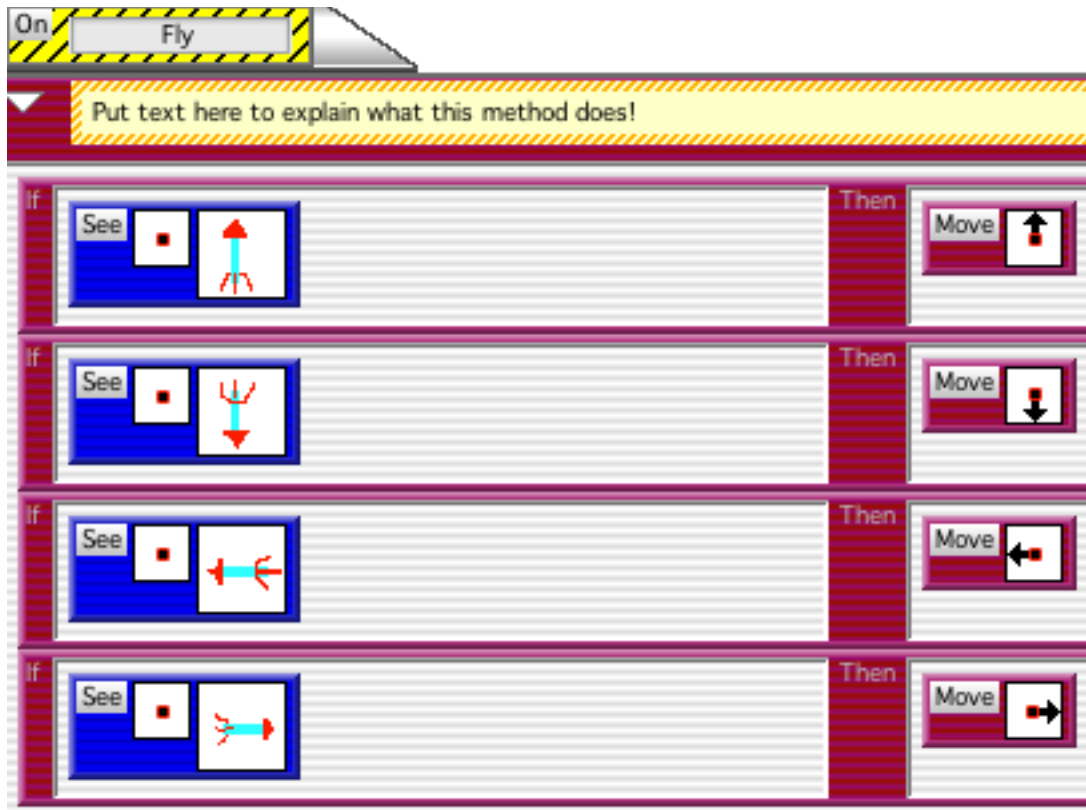
removing the move action and adding this action: which means "make **me** do the method named **Fly**".

- Add a timer condition to the if side of this rule to control how rapidly the arrow moves. Make it move slowly enough to be visible! The rule should look as follows:



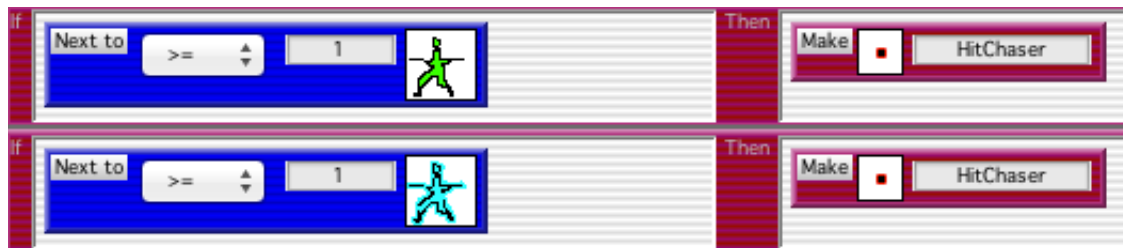
Create the fly method and add 4 rules to it so that the ice arrow continues to move in the direction its point faces:

- Click on the New Method button at the bottom of the Ice Arrow's rule window.
- Edit the method name so it exactly matches the name in the Make action.
- Create the first move rule by adding a condition that checks what the ice arrow looks like and adding an action that makes it move in the matching direction. For example, the up-arrow should move up.
- Duplicate the first rule 3 times and edit each of these rules so that the ice arrow can move in the other 3 directions.



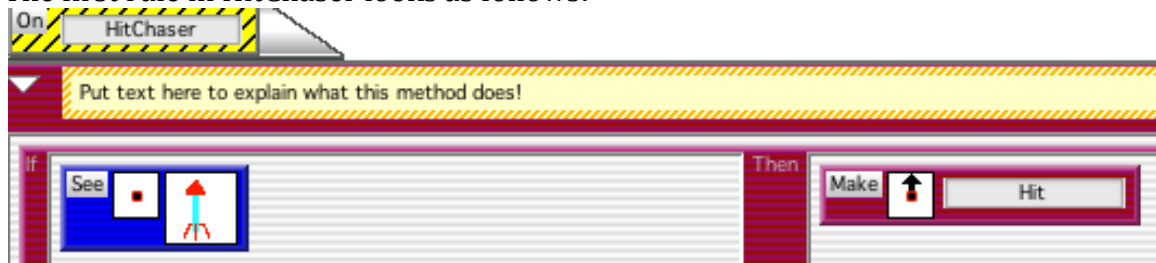
Edit the ice arrow rules so that each Ice Arrow depiction can freeze or unfreeze a Chaser:

- Change the hit rules in the while running method so that you have two rules which detect when an ice arrow is near an unfrozen Chaser and when it is near a frozen Chaser. In each rule, use the Make action to make the ice arrow do a new method, HitChaser. The rules should look as follows:



- The HitChaser method rules will decide which way the ice arrow is facing and where the Chaser is relative to the ice arrow. Once the HitChaser rule has checked the ice arrow depiction, it can send a Hit message to the Chaser in front of the ice arrow point so that it either freezes or unfreezes, depending on its state.

- The first rule in HitChaser looks as follows:




- Make 3 more rules in the HitChaser method to detect the other 3 ice arrow depictions and send messages to Chasers below, left or right of the ice arrow.
- Why did we create the HitChaser method? The logic of what happens when an ice arrow is next to a Chaser is the same whether the Chaser is frozen or unfrozen so we can create a single set of rules in HitChaser that will cover all 4 ice arrow depiction possibilities. The 2 rules in the main ice arrow while-running method make sure that both frozen and unfrozen Chasers can be hit by ice arrows.

Testing

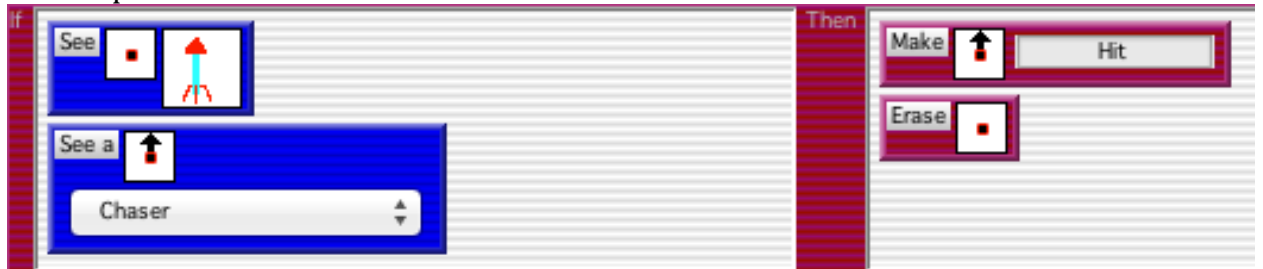
- If the agents' behavior does not match the changes you have made, click on each agent's apply button.
- Test that your Traveler can fire arrows in all 4 directions.
- Do ice arrows build up on the edges of your worksheet? Rearrange your walls to absorb them.
- If you have an arrow that does not move, check that the depiction in the New action in the Traveler's rules matches the depiction in the Move action in the ice arrow's rules.
- If you get error messages from other agents on the worksheet saying that they do not know how to respond to a "Hit" message, you must change the HitChaser rules to test that the ice arrow is about to hit a Chaser and not some other agent. Add a condition that checks for the Chaser agent by name rather than image to **each** HitChaser rule. Here is the upwards-facing ice arrow rule:



- If your Chaser is not frozen by a direct hit, the problem may be that the ice arrow is next to the Chaser and unfreezes it immediately after freezing it. How can you guarantee that an ice arrow will not cause the Chaser to change rapidly back to

unfrozen? Add an "Erase me" action, , to each of the HitChaser rules so that the ice arrow sends a Hit message to the Chaser and instantly disappears. Here

is a completed rule from HitChaser:



- Now test that your Traveler can shoot in all 4 directions and can freeze and unfreeze a Chaser with all 4 ice arrows. You may need to move the Traveler and the Chaser into position in order to test each direction.

Note: You may compare your completed journey game with the journey game with ice arrows saved with this activity.