Code Snippets

A summary of useful code snippets to be used as a reference for the student and teacher.

Created by: Susan Miller, University of Colorado, School of Education

This curriculum has been designed as part of the Scalable Games Design project. It was created using portions of prior work completed by Fred Gluck and Cathy Brand.

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.
Absorb

Absorb enables the agent to delete itself when it sees another agent. Example: "I want the cars moving to the right to disappear in the tunnel it sees to the right."

Collision

Collisions deal with the event wherein two agents run into each other. An example of the collision pattern occurs in Frogger, when a moving truck hits a frog. The truck wants to continue driving to the right but the frog is in the way. The live frog needs to be replaced by a dead frog when this happens.
User Control

User Control enables the agent to be moved using keyboard commands.

---

Generate

Generate enables new agents to be created. As shown in the example below, to simulate real world examples, a probability and timer may be included to make the scenario more realistic.
**Diffusion**

Diffusion is the process of spreading an agent attribute through a worksheet. For example, we might want to spread a smell through the worksheet. There are two different agents that will need code: the stinky agent (who gives off the scent) and the receiving agent (the ground that diffuses the scent). Once the scent is diffused, then Hill Climbing allows an agent to follow the scent.

**Code to create the smell by the stinky agent:**
Place this code at the END of the rules for the agent.

**Code to diffuse the smell by the ground agent:**

The “set” action sets each ground agent’s attribute “s” to the average of the attributes in the agents above, below, and on each side:

\[ s = 0.25 \times (s[\text{up}]+s[\text{down}]+s[\text{right}]+s[\text{left}]) \]

**Why do we multiply by 0.25?**
When you find the average of a set of numbers, you add them up and divide by the number of numbers.

In this case, dividing by 4 is the same as multiplying by 0.25.
Hill Climbing (Seeking)

Hill Climbing is a method of simulating an agent following a ‘scent’ propagated by diffusion. In Hill Climbing, the Chaser will periodically (Once every ___ seconds) run a method that instructs him to ‘smell’ up, down, left and right. This algorithm compares the scent attribute value in those four locations, and goes in the direction of the highest value.

This process of checking the four directions can be replaced by a single command, as shown below:
Broadcast and Polling

Broadcast and Polling is a CTP that directs the Controller agent to count the number of remaining agents (such as goals) on a worksheet. This is similar to a teacher who asks her students ‘who is still working?’ Students raise their hand, and the teacher counts the students. Work continues (with the teacher periodically re-asking the question) until all students are finished (the number is zero).

**Code for Controller:**

![Behavior: Controller](image)

**Code for agent being counter (goal)**

![Behavior: Goal](image)
Increment Numbers

Incrementing numbers is not a CTP, but it is an important skill for game design that enables one to keep score within a game by counting up incrementally.

First – tell the controller to make the number increment. In this case, we used a special method using the tools...the method is called when the user clicks on the controller using the finger tool.

Then – tell the number HOW to increment. The first three are shown...the rest must be added.

Note – there are special rules for what happens when there is a nine in the one’s digit!
Push

Push is a CTP that directs an agent to push an item one step forward, and then also move one step forward with that item.

Code to tell the pusher what to do when the arrow keys are pressed

![Behavior: Pusher](image1)

Code for Pusher to Move when he sees floor, destination or box

![Behavior: Pusher](image2)
Code Snippets (Continued)

Code for Box when pushed

[Diagram of Box behavior]

Code for Pusher to move after box is moved.

[Diagram of Pusher behavior]
Transport

Transport is a CTP that enables one agent to carry another agent. In this example, the turtle carries the frog.