Homework 13 (Chapter 14)

The InClass is to be demonstrated (or mailed) to your group leader. To make best use of the time you have in labs, be sure you have
(1) read the assignment before lab
(2) read appropriate background material before lab

The Homework is to be submitted via Eagle. It is to be individual work. You may talk to other students in the course about your design and to get ideas, but you are to write the complete Alice programs by yourself. You may receive help from the Professor, CS Tutors, UTF's or TA. In your comments, identify (by name) all those who helped you. Failure to do so is considered cheating.

In most cases, it will not be possible to finish the homework during labs. Plan on spending several hours per week outside of class to get the homework done.

In class (5 points)
Jumping fish: Create a sea world with an island (Environments/Island) and a fish (Animals/fish) (partially out of the water). The fish should swim around the island (turn "as seen by") until you press the up arrow. Then the fish should jump out of the water. Try using one of the skies (Environments/Skies).

Point Distribution
- (1 point) Setup with sky
- (3 points) Fish jump when press the up arrow
- (1 point) Camera move with mouse

Hints: I made the jump duration a parameter to the jump method so I could easily experiment with different times.
Homework 13  (due Friday, midnight)

In this project, you are to create a turtle motion controller to help the turtle (Animals) perform exercises. Create a world that contains only a turtle. Create motion control for the turtle:

- oneStep: causes the turtle to move forward one step. His legs should move while he is taking that one step.
- turnaround: turns the turtle 180 degrees. The turtle should be walking while he turns
- turnLeft/turnRight: turns the turtle left/right while he is walking
- hide: allows the turtle to hide in his shell (head and legs disappear; shell drops to the ground)
- reappear: allows the turtle to reappear from his shell (do NOT assume that the turtle is already hidden)
- talk: The turtle will look at the camera and say something.
- Object parameters (optional): Write the methods to accept an object parameter. Then pass in one of two turtles in to the method. Use a "billboard" to tell the user what keys he/she will use to control the second turtle.

Create keyboard controls

1. (3 points) When up arrow key is pressed, the turtle is to walk forward
When the down arrow key is pressed, the turtle is to turn around.
When the left arrow key is pressed, the turtle is to turn left.
When the right arrow key is pressed, the turtle is to turn right.
2. (5 points) When the letter "H" is pressed, the turtle is to hide in its shell.
When the letter "R" is pressed, the turtle is to reappear from his shell. (Use the same method with a different parameter value.)
3. (1 point) When the letter "T" is pressed, the turtle is to talk.
4. (3 points) When the letter "S" is pressed, the turtle is to do some other action of your choice. Be creative.
5. (3 points) Use of class variables.
6. (BONUS 1 point) Write the methods to accept an object parameter. Then pass in one of two turtles in to the method. Use a "billboard" to tell the user what keys he/she will use to control the second turtle. Test the turtle motion control system by running your world and trying all the interactions at least once. Be sure to hide the turtle only when he is already out of his shell and have him reappear only if he is already hiding.

Hints:

1. Instead of having a different method for left, right, and turn around. Use one method (changeDirection) and give it different arguments. Turning right is a left turn of -.25 revolutions. Being able to use the same code for multiple purposes is called "reuse".

2. This negative number trick also works with the hide/unhide. If you move “up” a negative number, it moves down. I made a separate routine to “shrink” that had a parameter “howMuch”. If I wanted the head to move farther than the legs, I multiplied “howMuch” by a factor. I then called the same shrink method from “hide” or “unhide”. This made experimentation with amounts much easier and forced the hide and unhide to do exactly the opposite thing.
3. Do use a “class” Boolean variable called “isHidden”. You will use this to keep track of whether or not the turtle is already hiding inside his shell. Otherwise, if you ask a hidden turtle to hide, you’ll have your turtle’s legs walking on top of his shell.