

HW6—Fish Tank

15 points

- Assignment:** Write a Python program that draws fish swimming around in a fish tank, as described below.
- Due:** Friday 10/4, 5PM
- Turn in:** Submit your source file to BBV. If you used images other than the ones I supplied, submit them too.

For this assignment, you'll be writing a program that displays fish swimming in a fish tank, as shown here.

There are three images on BBV in this HW area: `fishLeft.gif`, `fishRight.gif`, and `tank.gif`. (Feel free to substitute your own pictures, but they must be .gif files.) These must be in the same folder as your source file.

You can read the images into Python variables like the clock demo did.

Add the background (`tank.gif`) to the canvas, like the clock demo did with the clock face.

We want to be able to add any number of fish, so your “Add New Fish” button should call a function

that creates a fish, much like `startClock()`, but we won't call it directly from the main program: let the button call your `createFish()` function. Your `createFish()` function should choose the fish initial x and y locations, the dx and dy velocity, and create the fish images (one left, one right). All of this info should be passed to a second function called something like `updateFish()`. The `updateFish` function will be responsible for moving the fish (adding dx, dy to x and y), setting the fish images' locations, and scheduling itself to be called again by `canvas.after()`.

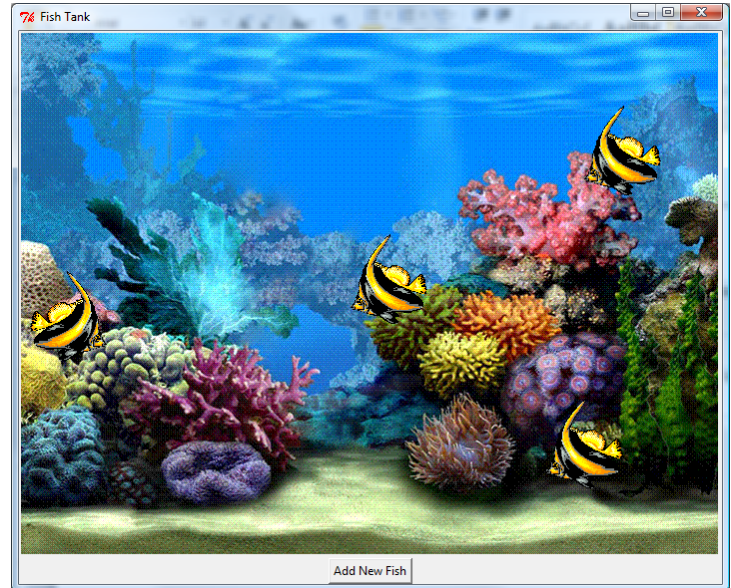
Your `updateFish()` function is where most of the interesting work will go. Each time `updateFish()` is called, update the x,y coordinate variables, and move the fish on the canvas by updating its coordinates, and scheduling `updateFish()` to be called again after 1/10 second or so.

The initial x and y coordinates of the fish should be selected at random (use `random.randrange()`) somewhere inside the tank. Also, select dx and dy at random between -3 and 3 inclusive.

For starters, get the fish to appear (use either the left or right facing fish) and swim around. At this point, the fish will disappear off of the canvas as it approaches an edge. But, at least you got to see a fish swim!

Next: in `updateFish()`, add code that detects whether the x coordinate reaches the left or right edge of your canvas. If so, negate dx so the fish swims the other way. Likewise, if y reaches the top or bottom edge, negate dy.

At this point, you should have a fish swimming around, bouncing off of the tank edges. But unfortunately it always faces the same direction, so sometimes it looks like the fish is swimming backwards!



To get the fish to face the correct direction, you'll really create *two* fish: one using the left facing image and one using the right facing image. When it's time to update the fish coordinates, check the sign of dx to see which way the fish should face. If dx is positive, move the right facing fish to the calculated x, y position and move the left facing fish off of the canvas entirely (like to location $(-1000, -1000)$). If dx is negative, do the opposite.

It should be possible to press the "Add New Fish" button as many times as you wish, each should add an independently moving fish.