1) Setting up the scene
- Use the seafloor for the ground
- Drag in these objects:
  - Swimmer (Fish): blueTang, clownFish, Shark
  - Swimmer(MarineMammal): dolphin
  - Props: adelaideBust
- See next slide on where to place them

Placement of objects
- Put the shark in the back.
- Make sure the shark and two little fish are all about the same height above the ground.

Placement of objects (2)
- The dolphin and headbust don’t matter much, turn them both invisible. We will use them later.
The story

• The two little fish will move around randomly all the time.
• The shark is hungry, he will move towards one of the little fish until he gets to it and then eat it. He will do the same with the other little fish.
• After the shark has eaten, we will slowly turn him into a dolphin, and then into a blueTang, and then into the adelaideBust

2) Write a Fish procedure to randomly move a fish one unit

• No loop in this code, just move the fish once
• You should generate a random number and then based on its value
  – 1/3 of the time have the fish move forward
  – 1/3 of the time have the fish move right
  – 1/3 of the time have the fish move left
• When the fish moves, it should randomly move some amount between 0.0 and 1.0
• Make this move happen quickly, 0.25 sec

3) Create a SceneActivationListener

• Create a new SceneActivationListener event. You should already have one such event that calls myFirstMethod
• In this new event, as long as at least one fish is visible, they should both move randomly (call the procedure you just wrote for each of them).
• You should use a while loop that repeatedly moves each fish once, and stops when both fish are invisible
• Run your program!

4) The story starts

• In myFirstMethod, we want to move the camera so it can see the action. Have the camera do the following:
  – moveAndOrientToAGoodVantagePoint of the shark (it will get a good look at the shark)
  – Then move the camera back 12 units
  – Then move the camera down 1 unit (you can adjust these if you think they need it)
  – Then glue the camera to the shark so when the shark moves, the camera also moves. (set the vehicle property)
5) Write a **Shark Procedure** called `catchfish`
   - There should be one parameter of type swimmer named `fishy`
   - Use a while loop to have the shark repeatedly move toward the fish (0.5), and repeatedly turn to face the fish. (these should happen fast, 0.25 sec each)
   - The shark should stop when its mouth is close to the fish. At that point open the mouth, and eat the fish (it should disappear), and close the mouth

6) Add to the story in `myFirstMethod`
   - The shark should catch and eat the blueTang fish and then the clownfish
   - Be sure to call the `catchfish` procedure

---

Write a **Scene procedure** called `ChangeInto`

- This procedure will take two objects and have one slowly turn into the other object.
- This procedure should have **two parameters** both of type `SjointedModel` called `critter1` and `critter2`
- We want `critter1` to turn into `critter2` and `critter2` should already be invisible
- First move `critter2` into `critter1` and have it face the same direction as `critter1`.

Write a **Scene procedure** called `ChangeInto` (cont)

- Use a while loop to repeatedly make `critter1` more faded (by 0.1) and make `critter2` more visible (by 0.1) until `critter1` is invisible (and `critter2` should thus be completely visible). This should happen fast! In 0.1 seconds with each change
- **CAUTION:** When you add 0.1 the computer can’t represent 0.1 exactly, so it might actually be represented as 0.100034. So if you add ten of them together, it is more than 1.0, but opacity can only be at most 1.0
Write a **Scene procedure** called *ChangeInto* (cont 2)

- To make sure you don’t go over 1.0, use the **min** function. That is for critter2 which is becoming visible, always assign it to the smaller of 1.0 and the number you are adding to. That way if that number gets too big, you will just use 1.0.
- For example, .100034 added 10 times is 1.00034, which is greater than 1.0. You cannot set the opacity to 1.00034! So set it to 1.0 in this case.

Write a **Scene procedure** called *ChangeInto* (cont 3)

- To make sure you don’t go under 0.0, use the **max** function. That is for critter1 which is becoming invisible, always assign it to the larger of 0.0 and the number you are subtracting from. That way if that number gets too small, you will just use 0.0 for the opacity.
- You can find **min** and **max** under **math**

Finish the story in myFirstMethod

- After the shark eats the two fish, have the camera zoom in toward the shark (10 units)
- Have the shark “changeInto” the dolphin
- Have the dolphin “changeInto” the blueTang
- Have the blueTang “changeInto” the adelaideBust
- **NOTE: the fading doesn’t always work too well, may look like it happens instantly.**