1) Setting up the scene

- Add in any ground cover, maybe ice?
- Drag in these objects so they are standing in a line all facing front
  - Biped: yeti, yetiBaby
  - Flyer: Penguin, Seagull, Phoenix
  - Quadruped: from myClasses: Your Dalmation who knows how to jump
- See next slide on where to place them

Placement of objects
In a line all facing front
- L to R: phoenix, seagull, Dalmatian, penguin, yetiBaby, yeti
- Check side view!

Move the penguin up 3 units
- Make sure you can still see the penguin, you may need to move the camera back.
- Use a one shot to do this.
2) Add in object markers so you can easily move the Yeti to saved positions
   - Go to scene setup
   - Select Yeti (object markers are for Yeti)
   - Add 3 Object Markers

![Image of Yeti with object markers]

Spread out the Yeti object markers
   - Put two of them spread out behind the creatures
   - Put the third one by the yeti

![Image of Yeti with object markers spread out]

3) Write a **Flyer** procedure called `flapWings`
   - Have both wings **turn** at the same time and then both turn at the same time back to where they were, just one flap of the wings.
   - This procedure does not need any parameters
   - Test out your procedure on the Phoenix and the Seagull

4) Write a **Flyer** procedure called `flyAround`
   - This procedure should have one parameter, of type `SJointedModel`, which represents the object to fly around
   - This procedure should move the flyer up 3 meters, then have it circle the penguin **twice** while flapping its wings **8 times** total (**4 times** in each loop) (be sure to call `flapWings`), and then move back down to where it started. Set the timing so it looks good!
   - Test it out!
5) Add code to myFirstMethod

- Here is the first part of myFirstMethod
  - Have the seagull *fly around* the penguin.
  - Then have the phoenix *fly around* the penguin
- Be sure to use the procedure you wrote!

6) Write the **Dalmatian function** named *distanceToJumpOver2*

- The Dalmatian needs to know **how far** to jump if it wants to jump over two objects.
- The function should return the distance of the object that is further away. What type should the return value be?
- Add in two object parameters so that the Dalmatian can jump over any of the creatures we have added to the world. What type should they be?

7) Add the next part in myFirstMethod

- Have the Dalmatian turn to face the seagull and then jump over both the seagull and the phoenix at the same time. Be sure to use the function you just wrote to calculate how far to jump and add a little bit to that to go past them
- For how high to jump, just have the Dalmatian jump its height.

8) Add a **property** to **yetiBaby** named *colorName* of type *string*

- We will give the yetiBaby a property that will store the name of its color, that is a string
- For YetiBaby, click on **add property**, name it colorName and set the type to TextString
When you add the property you get a procedure and a function for it

- New Procedure automatically added:
  - setColorName
- New Function automatically added:
  - getColorName

9) Write a YetiBaby Function named changeColor

- We want the YetiBaby to turn a different color based on a number we pass in
- The function will just calculate the color to change to based on the number. If the number is 1, blue. If 2, green. If 3, yellow. If any other number, red
- This function should return the color to paint the YetiBaby
- Add one parameter that is a Whole number

Also must update the colorName property

- Once you determine what color you want to paint the YetiBaby, you should reset the colorName property to a string that is the name of that color.
- You must do this before you return the color.
- Once you return from a function you leave it
- For example, for number 3, set colorName to “yellow” and then return the color yellow.

Finish the story in myFirstMethod

- Create a number variable named countValue that is set to 1
- Repeat the following 4 times (use a loop)
  - The yetiBaby should change its color based on the value of countValue
  - The yetiBaby should say “I’m the color _____” (fill in the color, use colorName to get the name of the color)
  - Update countValue by 1.
- Move the YETI to the two object markers behind, and then back to where the YETI was.
Last step

- When your program works, then make the penguin permanently invisible!
- Now play your world!

- What would happen if count loop is done 6 times?