Iffy Dogs Program

This lab assignment exercises the use of if instructions to make choices. It is extremely important that as you create the five objects in the scene below. Please select object names as follows:

- **boy** -- the left teen
- **girl** -- the right teen
- **wolf** -- the wolf
- **poodle** -- the poodle on the left
- **dalmatian** -- the dalmatian on the right

---

**Part 1**

Begin your program by dropping an if instruction, then complete the condition with a `getBooleanFromUser` from the wolf function menu. Your instruction should look like the following:

```plaintext
if (wolf) getBooleanFromUser("Turn toward girl?")
```

When you execute this program, a pop-up window will appear with the message "Turn toward girl?" and true/false buttons. The if instruction condition will take on the value from the user’s button choice. Next, complete the then-clause so that the wolf turns to face the girl and the else clause so the wolf turns to face the boy, as shown to the right:

Now modify your program so that regardless of which person the wolf faces, that same person jumps up and down. Add an instruction to the bottom of your program so that regardless of which person jumps, the wolf says "Grrr!" Run your program a couple of times to be sure it works properly when the user (you) selects true and also when the user selects false.
Append the following code to your program. Before explaining the precise programming steps, the story behind this code is that each person is trying to select a dog. The boy wants to call his dog *Fido*, while the girl prefers the name *Rover*. The boy and girl decide to allow the dogs decide who goes with which owner. These dogs are smart, so they know that if they both select the same person there will have to compete for food with another dog. Therefore, each dog will choose a different owner.

**Programming Steps**

1) **Using** `do together` **cause** boy **to say** "Here, Fido!" **and** girl **to say** "Here, Rover!" **simultaneously.**
   (Insert a *do together* tile, then insert your two say instructions inside.)
2) **Create/declare** a variable named *Fido*. Make this the first instruction of your program. This variable must be of type Quadruped and initialized to one of the two dogs (either poodle or dalmatian).
3) Declare a second variable, named *Rover*, of type Quadruped and assign it a dog (either poodle or dalmatian).
4) **Append** an *if* instruction to the bottom of your program. Change the *if* condition to `nextRandomBoolean` from the pull-down menu. (This causes the condition for the *if* instruction to randomly choose between true and false).
5) **Insert** assignment instructions into the *if* instruction clauses so that *rover* will always be assigned one of the dogs and *fido* will always be assigned the other, but this assignment is reversed in the else clause and the then clause. This is where the dogs make a choice. When the poodle chooses to be *fido*, then the Dalmatian chooses *rover*, and vice versa (Try a few options and if you get stuck, ask for assistance.)
6) **Following** the *if* instruction from Step 6, cause *Fido* to turn to face boy then move forward close to, but not quite touching the boy.
7) **Next,** *Rover* must turn to face the girl and move forward close to her.
8) **In** the end the two dogs should leap up, roll backward and back to earth simultaneously.

Run the program several times. If you did things properly, you should always see the two dogs move to opposite objects (one to boy and one to girl). You should also notice that the two dogs don’t always pick the same owner.

If **time permits:**
   - Cause the two dogs to do simultaneous backflips. This requires jumping up and down with a backward turn between. The trick is to get the backflips to occur in unison.

**Part 3 (as time permits)**

Modern programming languages must respond to **events**. An event occurs any time the user needs to interact with your program. User actions such as clicking a mouse button, dragging a slider or pressing a keyboard key are all events. In Alice events are processed by separate code often called “event listeners.” To create event listeners you must click on the *initializeEventListeners* tab instead of the *myFirstMethod* that we have used until now.

Create a new event listener by clicking on the *Add Event Listener* then select *keyboard / addKeyPressedListener*. You should see the image shown on the next page. Notice that the *drop statement here* space is where you insert instructions that you want to execute in response to a key pressed event (i.e., anytime the user strikes a keyboard key).
1) Now insert an if instruction in this event listener and complete the condition to say true.

2) Next notice the isKey tile near the right side of the above window. This is a logical condition, so drag and drop it in place of the true condition you inserted in step 1. When you drop the isKey tile you must select which key to test for. Please select arrows (left, right, up, down) then LEFT from the menus.

3) Now insert an instruction to cause the wolf to move to the left in your then clause. The picture below illustrates:

Run your program and click the left arrow key to see if it behaves as you expect.

4) Now modify this program so that the wolf first turns to the left and then moves forward.

5) Using this same event listener we can check for other keystrokes by adding to the if instruction. See if you can make the wolf move to the right when the RIGHT key is struck ... move forward when the UP arrow is struck. ...roll over when the R key is struck.