1) Setting up the scene

- Add in any ground cover, maybe sand
- Drag in these objects:
  - Biped: Hare, Bunny, Alice
  - Prop: CoinStack (3 of them), SaguaroCactus
  - TextModel
- See next slide on where to place them

Placement of objects

- Place objects here. Note there are three coinstacks, put them anywhere.
- 3DText is set to hello and then make it invisible

Story

- When the hare is close to the cactus for the first time, the cactus tells the hare to pick up the three piles of coins and bring them to the cactus. Note the Hare must say: move me close to the cactus to start the game.
- The hare collects the three piles in its right hand, a counter counts the piles as they are picked up. The hare brings them to the cactus.
- When the hare is close to the cactus, the piles of coins are dropped to the ground, and the cactus says: *You have completed the mission*
Story continued

• The cactus also says: Get close to Alice and tell her you got the coins.
• **Once the mission is complete**, if the hare goes near Alice, then she says “yeah” and she and the bunny spin around once.
• This should now happen every time the hare is near Alice

Note – MyFirstMethod should be almost blank!

• This whole story is interactive!
• The only thing in MyFirstMethod is the hare telling you to move it to the cactus to start the game.

2) Add in a ModelManipulation listener

• Try moving the objects around (back and forth, up and down and turn them).

For Steps 3, 4, and 5 on next slides

![Diagram with step 3, 4, and 5 markers]
3) Setup the counter

- The 3DTextString will display the counter. But it is a string, so you will also need an integer property to store the current count.
- Add a **property to the textModel object** called coinCounter that is a wholeNumber.
- Initialize it to 0.
- We will update this number and then display it with the visual 3D textModel.

4) Write a textModel procedure setupCounter

- It should set the coinCounter to the number 0.
- It should set the 3D text value (use setValue) to the value of countCounter. (first set to the string “” (empty string) and then add to that the number coinCounter – which turns that number into a string so it can be displayed.)

5) Write the textModel procedure updateCounter

- This should update the numeric counter, by adding one to it.
- Then it should set the text that goes with the 3D text to the value of this number. (that will make the updated number display).
6) Write SCENE procedure collectCoins to collect one stack of coins

- Create a parameter of type Smodel (this is for all the coin stacks) called coins
- If the coins are completely visible (opacity is 1) then
  - Move the coins to the hare's rightWrist and glue them to the wrist
  - Set their opacity to .9 (so we don’t pick them up again)
  - Update the counter by calling updateCounter

7) Add in another SCENE property called “state” of type textString

- We will use the state property to help guard events. It will start with the value “start”, then we will change it to “run” while you are collecting coins and then “finish”
- This will help us guard the event listeners to only work at certain times

8) Now let’s create the rest of the events – create pointOfViewChangeListener

- This event is under position/orientation. For the custom array put in just the hare
- If the hare is close to the cactus and the state is “start” then
  - Cactus should tell the hare to pick up the three piles of coins and to bring them to the cactus
  - Call setupCounter
  - Set the state property to “run” (that means this part of the event only happens once)

8) Continue the event with another if

- If the hare is close to the cactus and the state is in the “run” state and all three stacks of coins have been picked up (that is coinCounter should be 3) then
  - Put the coin stacks on the ground
  - Unglue the coin stacks
  - Set the state to “finish” (so this part of the event cannot happen again)
9) Create another `pointOfViewChangeListener`:
   - This one is also for the hare.
   - If the state is “run” (only pick up coins if the counter has been started) and the hare is close to any of the coinstacks, then call `collectCoins` with that coinStack (note you will need three ifs – one for each coin stack).

10) Create one more `pointOfViewChangeListener` for the hare:
   - If the hare is close to Alice and the game is over (that is the state is “finish”) then Alice should say “Yeah” or something like that, and both Alice and the bunny should turn all the way around.