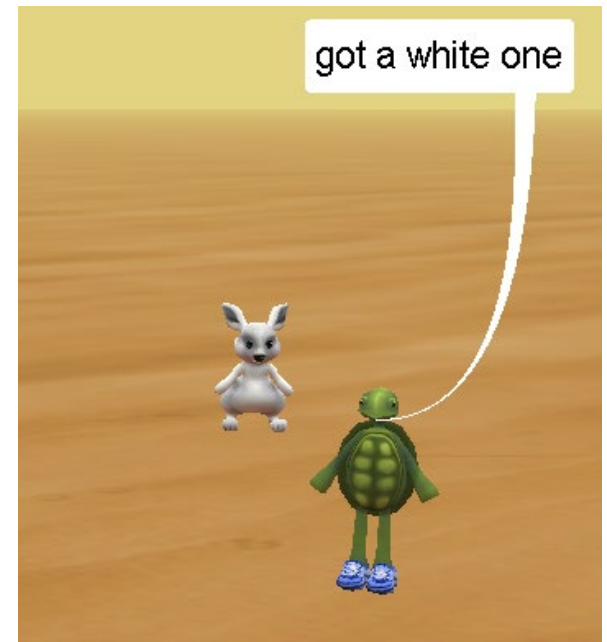


CompSci 94

Classwork: Mouse Event

October 28, 2021



Prof. Susan Rodger

1) Setting up the scene

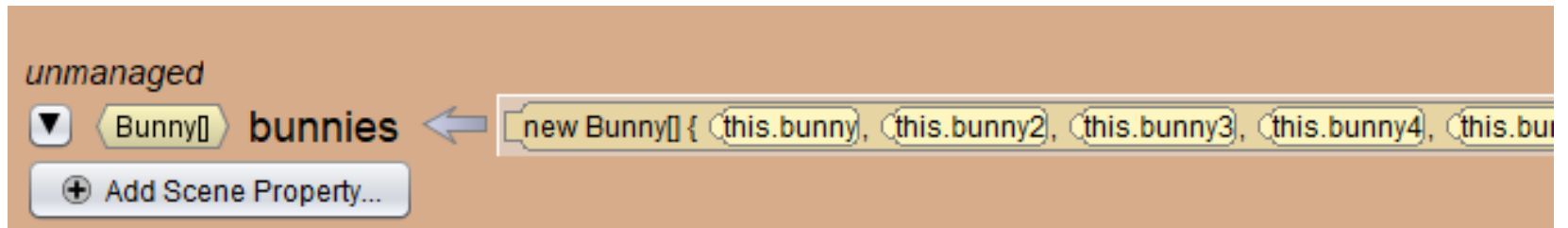
- Add in any ground cover, I used sand
- Drag in these objects so they are in positions similar to the picture on the next page
 - Biped: 9 bunnies, 1 tortoise
 - Paint the bunny named bunny, the first bunny created, gray, and paint bunny2 brown
 - Put the gray bunny on the left and the brown bunny on the right and spread the other bunnies out with one in the middle
 - Put the tortoise in the front right of the screen, with no bunnies behind it or to the right of it.
- See the picture on the next slide.

1) Setting up the scene (cont)



1) Setting up scene (cont)

- Create a Scene property that is an array named bunnies, and put all 9 bunnies in the array. (not all shown in the array below)



2) The game

- We will create a game where the tortoise will tell the player to click on the tortoise to start the game. The tortoise then will tell the player to click on the bunnies. Then the bunnies will all go below ground and one at a time randomly move up above ground and then back down. The gray and brown bunny always pop up in a different place. When a player clicks on a bunny the tortoise says “got a X one”, where X is its color. This game never stops.
- Follow the steps that follow to build this game.

3) Setup the centered bunny

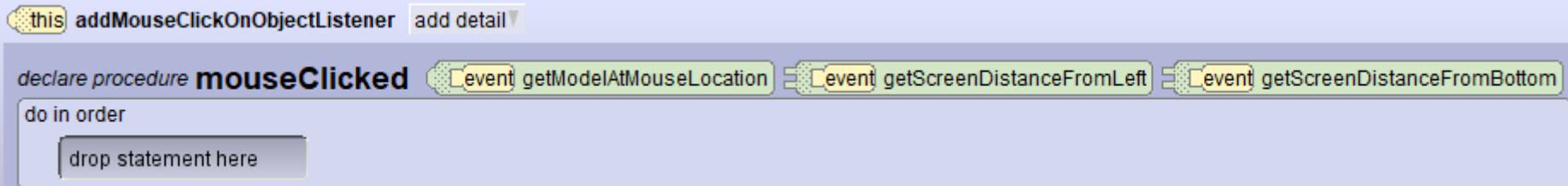
- Go to scene setup. Click on the bunny in the center.
- Add an object marker for it
- Use a one-shot to move it down one.
- Move the object marker down to it with the controls.
- Move that bunny back up above ground.
- You'll use the object marker to always make sure the white and gray bunnies start in the center before they move to a random location

4) Tortoise event, **idea** for this event only happening once

- When the player clicks on the tortoise, the game will start. We only want this to happen once, so we will guard this event by making the tortoise change its opacity a little when the tortoise is clicked on, to 0.9. Then this event will only happen if the tortoise's opacity is greater than that. So the event can only happen once.
- Follow the steps on the next slide to create this event.

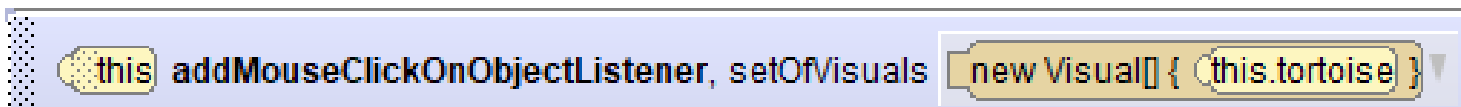
4) Create Tortoise event

- Click on the **initializeEventListeners** tab, and then click on **AddEventListener**
 - Select mouse, and then add **MouseClickedOnObjectListener**



```
this addMouseClickedOnObjectListener add detail  
declare procedure mouseClicked event getModelAtMouseLocation event getScreenDistanceFromLeft event getScreenDistanceFromBottom  
do in order  
drop statement here
```

- This event is **ONLY** for clicking on the tortoise
- At the top click on “add detail” and select **setOfVisuals**, and **customArray**, and then create an array with just the tortoise.



```
this addMouseClickedOnObjectListener, setOfVisuals new Visual[] { this.tortoise }
```


4) Tortoise event (cont)

- Add code to do the following to the event for clicking on the tortoise (in this order):
 - If the tortoise has opacity greater than 0.95
 - At the same time move all the bunnies down 1 (to below ground)
 - Then change the opacity of the tortoise to 0.9 (you won't notice). That makes this event only happen once
 - Then have the tortoise say “Try to click on the bunnies”
- Run your game. Click on the tortoise and the bunnies should move down.

5) Write **bunny** procedure named **moveRandom**

- This procedure is only for the gray and brown bunnies, to always start in the center and then move a random direction.
- Add in a do in order
- You'll want to have the bunny move and orient to the object Marker that you created earlier. Only Alice won't let you access it. So you will need to create a parameter of type SThingMarker named bunnyMarker.
- Have the bunny moveAndOrientTo the bunny Marker. This should happen instantly, in 0 secs.

5) bunny procedure **moveRandom** (cont)

- Next, at the same time, have the bunny randomly move forward from -3 to 3. If it generates a negative number then it will move backwards.
- At the same time the bunny moves forward, have the bunny move right from -3 to 3. If it generates a negative value then it will move the bunny left instead of right.
- That's it.
- You'll test this later, when you use it.

6) Write the game in myFirstMethod

- Add a doInOrder into myFirstMethod
- Have the tortoise say “Click on me to start the game”
- Add in a while **true** loop next and in the while:
 - You want one of the bunnies randomly to pop up. Generate a random index number and store it in a variable. There are 9 bunnies so this number should be from 0 to 8. Note this should be a whole number.
 - Have the bunny at the random index location move up 1 in 0.25 units, delay 0.5 and then move down in 0.25 sec.

7) Run your game.

- You will notice if you don't click on the tortoise right away, the bunnies will start moving. We need to add a pause before the while loop that plays the game. How could we pause the game?
- We know the game is ready when the tortoise changes its opacity to 0.9
- See next slide to see how to pause the game

8) Pause in myFirstMethod

- In myFirstMethod, right after the tortoise says to click on it, add a while loop.
 - While the tortoise has opacity 0.95 or greater, do nothing, the while loop body is empty!
 - This first while loop exits when you click on the tortoise.


```
declare procedure myFirstMethod
do in order
  do in order
    this.tortoise say "Click on me to start the game" add detail
    while this.tortoise getOpacity > 0.95 is true
      // wait for tortoise to be clicked on, meaning its opacity is set to 0.95
      loop
    while true is true
```

9) Now run the game

- The bunnies should not move until the tortoise is clicked on. Then the game should start.
- Try clicking on a bunny!
- At this point you don't know when you have clicked on a bunny. Let's fix that next.

10) Add an event for clicking on bunnies

- Click on the initializeEventListeners tab
- Add an event listener for clicking on mouse, addMouseClickedOnObjectListener.
- This event is for any bunny. Click on add detail, then setOfVisuals, and then select the bunnies array.



```
this.addMouseClickedOnObjectListener, setOfVisuals this.bunnies
```

- In the event, have the tortoise say “got a X one” where X is the paint color of that bunny

11) Run your game

- Click on the tortoise, and then try clicking on a bunny. The tortoise should tell you when you clicked on one.
- If it is too hard to click on a bunny, change the delay to longer when the bunny pops up.

12) Have the brown and gray bunnies move randomly

- Whenever the gray or brown bunny is randomly chosen, we want it to first move to a random spot before it pops up.
- To add this:
 - Click on the myFirstMethod tab.
 - Right before the bunny moves up, check to see if the bunny selected is the brown or gray bunny. If it is, then have it moveRandom first.

13) Now play the game

- Whenever the gray or brown bunny pops up it should pop up in a different place.



14) Guard clicking on bunnies

- You don't want the user to click on bunnies until they are underground. On the event where you click on bunnies add an if statement around all the code so the bunnies don't move up until after the tortoise says they can.
 - What condition do you need in the if?
- Play your game and click on a bunny before the tortoise says to. Nothing should happen yet.
- That's it!
- We will add more to this game next time!