Repetition everywhere – comparing *while* in a method and as an event

Comparison
- This tutorial will explain the difference between using “while” in a method and using “while” in an event (as a BDE).
- It will also explain the difference between the “when” event and the “BDE” while event and when to use each one.

Start with a new world
- Use any ground, I use the sand.
- Add in a bunny, chicken and cow. Put the chicken on the back right side, the bunny on the front left side and push the cow back far on the left side. The exact position is not important.

First Experiment with loop
- We will first experiment with using a loop to get the chicken to move over to the bunny to greet it.
- In the process we will teach the chicken to move forward while shaking its head.
- In myFirstMethod add in a Do In Order and have the bunny and chicken turn to face each other.
First Experiment with Loop (2)

- Next in the Do in order, drag in the “loop” tab from the bottom of the Alice window and select 3 times
- Put inside the loop a “Do in order” first, then inside this Do in order have the chicken move forward, then turn its neck a little, 0.1 revolutions each backward and forward. Make the turn happen faster by changing the duration on the turns to .25 seconds
- After the loop, add in the Chicken saying a greeting
- See the next slide to see the code.

You’re “my first method” should be:

Click “play” now

- The chicken moves forward three times shaking its head, but doesn’t get close to the bunny.
- The “loop” command is useful when you know how many times you want to repeat.
- In this case, we do not know how far the chicken is from the bunny, so loop does not help us much.

Change the loop count

- Try changing the loop to another value, what happens? Is it hard to determine what the correct value should be? What if the chicken gets moved to another position?
- Try changing the loop count to forever which is the choice “infinity times.
- When does the chicken say its greeting?
Infinite loops are dangerous!

- The code AFTER an infinite loop will never execute, because “infinity” is forever! The loop never ends.
- Change the loop count back to 3.

Then DISABLE the loop so it no longer executes by right clicking on the leftmost tab and select disable.

Use While loop when the loop should stop based on a condition

- Instead of using a loop that executes a fixed number of times, we would like the chicken to stop when it gets “close to” the bunny, about 2 meters in front of the bunny
- Drag in a while tab from the bottom and put it between the disabled loop and the Chicken say, and select true.

Move until Close to Bunny

- Have the chicken keep moving forward as long as the chicken’s distance to the bunny is greater than 2 meters.
- Under world functions find “a>b” and drop it in the true, select 1 and 2.
- Then under chicken functions drag over “chicken distance to” and drop over the 1, and select “bunny, the entire bunny”

Put the chicken movement in the while

- Copy the “do in order” inside the loop and put the copy inside the while

- Now Play and the chicken should stop in front of the bunny and say a greeting
Next we will show how to use a while in an event

- We will now reorganize the world to be all event driven.
- We don’t want myFirstMethod to run, but we want to leave the code there so you can compare it to the event while.
- Right click on the event “when the world starts Do: my first method” and choose disable
- If you can’t disable it, then delete the event

Create a BDE (while) event (B)egin, (D)uring, (E)nd

- In the event window, click on “create new event”, and select “while something is true”

- Resulting in the event:

Meaning of BDE

- Constantly while Alice is running it checks the condition of the BDE.
- If the condition is true then the following happens:
  - As soon as it becomes true, the Begin code executes, and then the During code executes.
  - As long as the condition stays true, the During code keeps repeating.
  - When the while condition becomes false, then the End code executes

Condition for BDE

- Put in the same condition “chicken distance to bunny is > 2”. Remember to drop in “a>b” first. The code then looks like:
Begin code:

- Since we want to put two lines of code in the Begin section, first put in a “Do in order”. Then add code to the the chicken and bunny to face each other.

During and End Code

- Copy the code from myFirstMethod and put in the During and End as shown below. You can make copies and drag them or use the clipboard.

Now Play

- Since the chicken is further than the 2 meters from the bunny (assuming you have the chicken back far), this event starts when Alice starts. You should see the chicken move within 2 meters of the bunny and then give a greeting.

Now lets add an event to move the animals

- Click on “create new event” and select “let the mouse move <objects>”
- You should see the event

  ![Let move Any Object](image)

- Click on “Any Object” and select “create new list”
- Type in the name “animals”, select object for type, click “new item” three times, and then replace the “none”s with chicken, bunny and cow. See the next slide.
Then click ok

Finished event

- Your event should now look like this:
  
  ![List of animals](image)

- This lets the mouse move only those objects in the animal list. You can see the list under world properties.

Now Play

- When you play your world the chicken moves to the bunny and greets it.
- Now click on the chicken or bunny and quickly move it back and let go. The event condition is true again so the event starts again.
- CAUTION. You need to move the chicken quickly. If you take too long then the chicken and bunny will turn to face but you will still be moving the chicken and when it starts moving forward it may not be facing the bunny. This could cause it to pass the bunny and move FOREVER.
- You could add a Wait 3 seconds in the BEGIN before the turn to face commands to give you more time to move it. Do this now. Then play.

Begin code with the wait 3 seconds

![Begin code](image)
Compare “when” event to “while” event

• We will create one more event to compare a “when” event to a “while” event
• A “when” event is like an “if” in that it only executes one time, it is not a loop. Whenever its condition becomes true its body is executed one time. It will only execute again if the condition becomes false, and true again.
• The “when” event is hidden, it is not in the menu.
• Click on “create new event” and select “while something is true”

Convert “While” BDE to “when” event

• Right click on the leftmost part of the while BDE, and select “change to”, then “when something becomes true”

• The event now looks like”

Finish When event

• The when event is useful for having one thing happen when something is true, such as having the cow say “Moo” whenever it is close to the chicken.
• Put in the condition “cow distance to chicken < 2” and have the cow say “Moo”. Remember to drag in a<b first.

Now Play

• The chicken goes to the bunny and greets it.
  (While BDE starting first time)
• If you move the cow close to the chicken, the cow says Moo. You have to move it away and back again and then this event is triggered again. (When event starting)
• If you move the chicken or bunny away (within 3 seconds), then the chicken goes to the bunny again. (While BDE event starting again)
Part 2 – Guarding the Event

- We may only want the event where the chicken greets the bunny to happen at certain times during the program.
- We will guard it by adding another condition to the BDE.
- First, add a square to the world from the Shapes folder and put the square in the top right corner.
- Change the color of the square to red using the square’s property.

Let’s add events to change the color of the square

- In the events window, create a new event, “when a key is typed”.
- Over “Any key” select “letters”, then “G”.
- Click on Square, properties.
- Then drag color into the “Nothing” part of the event and select “green”.

Add a similar event

- Now make a similar event for when you click on the letter R, the square sets its color to red.
- Play the world and press R and G.
- The square changes color slowly. To speed it up, click the “more” and change the duration to 0 seconds for both the R and G events.

Now guard the BDE event

- Let’s add an additional condition to the BDE so it doesn’t happen all the time. We’ll add that the square has to be green.
- From world, functions, drag “both a and b” to the while condition.
Now guard the BDE event (cont)

- Click on “square”, “properties”, and drag “color” into the “true”, and select “green”

Now Play the world

- Wait a few seconds, nothing happens
- Press the G key to turn the square to green
- Then the BDE starts and the chicken heads to the bunny. Let it reach the bunny.
- Then click and drag the chicken far away. The BDE kicks in again. Let the chicken reach the bunny.
- Press the R key to turn the square to red.
- Now drag the chicken far away. The BDE does not start since the flag is not green.

Review, Extras

- Review: The BDE event becomes true only when BOTH the chicken distance to the bunny is greater than 2 AND the square’s color is green. They both have to be true.
- Extras.
  - If you don’t like the square showing you can make it invisible and the world will still work the same way. (set’s the square’s “is Showing” property to “false”)
  - If you only want the BDE to happen once, then in the End part of the BDE, set the square’s color to red.