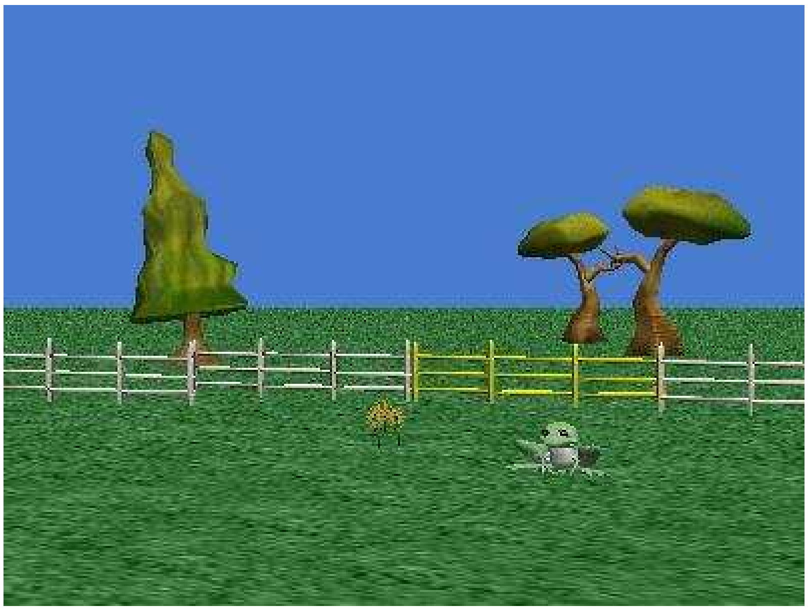
Introduction labb

Open Alice environment. If the tutorial is not starting automatically go to Help in the menu and start the tutorial. Follow the examples in Tutorial 1-4. Try to understand what happened and why when doing the instructions.
More about programming with Alice:
http://www.developer.com/java/other/article.php/3673761

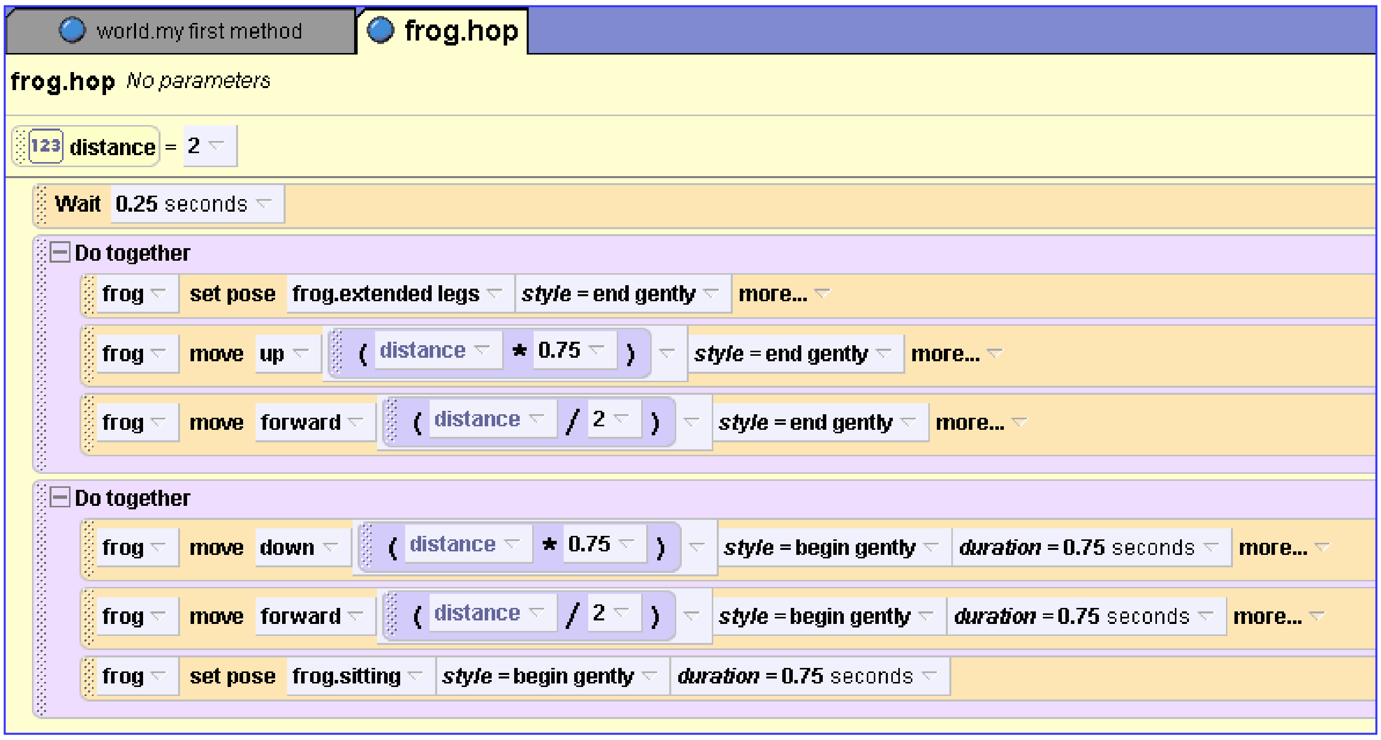
Exercise 1, create your own world

Create a word containing some tees, flowers, fence and a frog. Putt all objects together so the world will look (about) like bellow. Save your world. In the next exercise you will make the frog hop along the fence.



Exercise 2.

If you right click on the frog object you can see all the operations frog-object “can do”, we call that methods. Try some methods…
But, we want the frog to hop. Unfortunately you don’t have a ready implemented method hop. So, we need to create that method for the frog. Make sure the frog-object is in focus and click “Create new method” and name it “hop”.
(Se the example next page. You don’t need to do set pose)



```
world.my first method | frog.hop
frog.hop No parameters
123 distance = 2
  Wait 0.25 seconds
  Do together
    frog set pose frog.extended legs style = end gently more...
    frog move up ( distance * 0.75 ) style = end gently more...
    frog move forward ( distance / 2 ) style = end gently more...
  Do together
    frog move down ( distance * 0.75 ) style = begin gently duration = 0.75 seconds more...
    frog move forward ( distance / 2 ) style = begin gently duration = 0.75 seconds more...
    frog set pose frog.sitting style = begin gently duration = 0.75 seconds
```

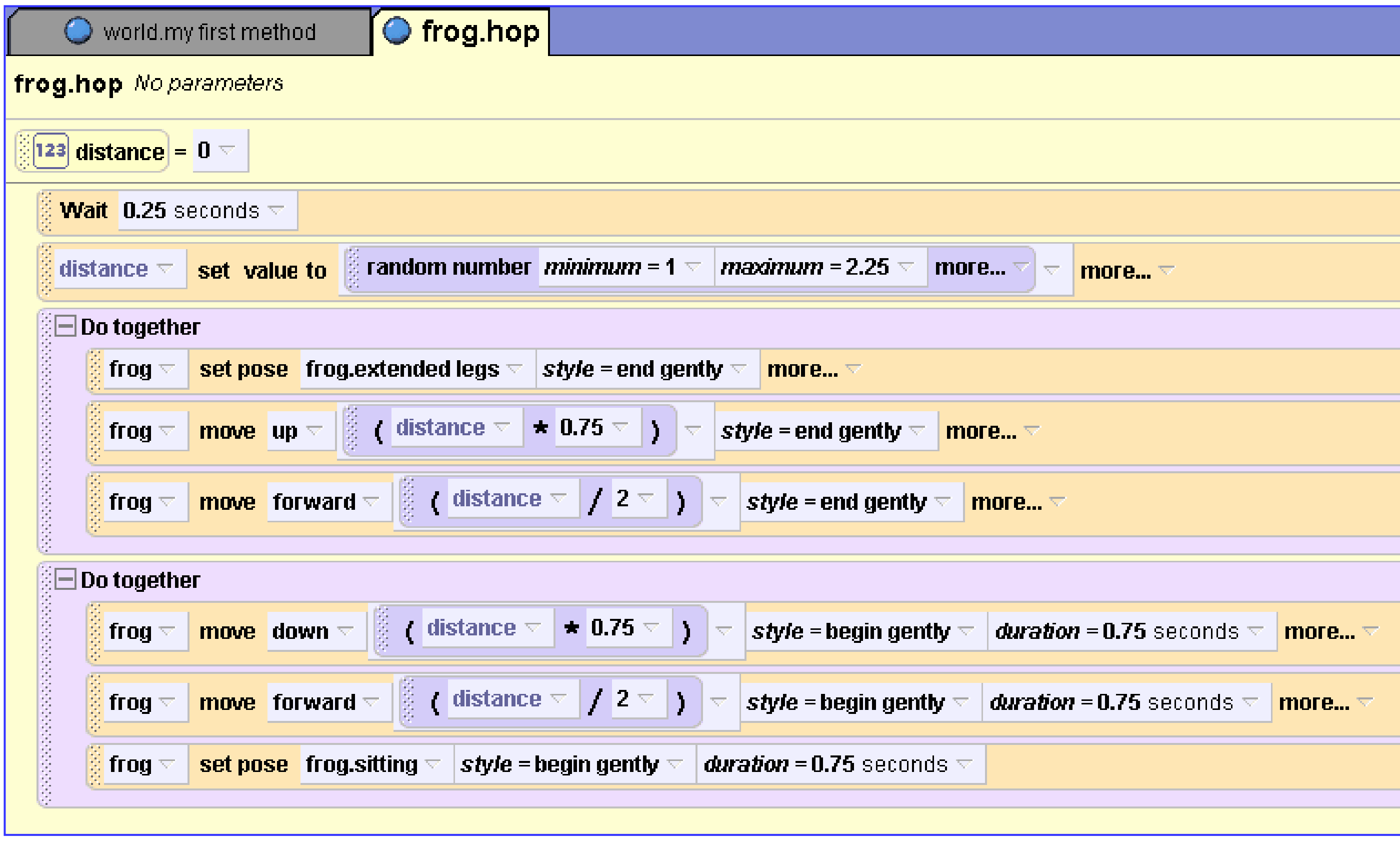
Now, go to world and “my first method”. Call the hop method 3 times. Run your program.

Exercise 3

Modify the world so the frog turns the hed to the left and a slight amount after each hop. For that, first you have to create a method called “slight_turn” who determinate the turn and executed. Call the method from the hop method.

Exercise 4

Add a new frog –object to the world. Change the propriety color for this frog object to red. Implement for this object a method called “hop_random”. This means that the distance to frog move forward will be determinate by a random value. See example below.
Call this method from the “my first method” and se how it works.



```
world.my first method | frog.hop
frog.hop No parameters
123 distance = 0
Wait 0.25 seconds
distance set value to random number minimum = 1 maximum = 2.25 more... more...
Do together
    frog set pose frog.extended legs style = end gently more...
    frog move up ( distance * 0.75 ) style = end gently more...
    frog move forward ( distance / 2 ) style = end gently more...
Do together
    frog move down ( distance * 0.75 ) style = begin gently duration = 0.75 seconds more...
    frog move forward ( distance / 2 ) style = begin gently duration = 0.75 seconds more...
    frog set pose frog.sitting style = begin gently duration = 0.75 seconds
```

Exercise 5

Implement a new method called “ frog_OS”. In this method call the hop method and the hop_random method 4 times each.
How it looks like. Witch frog wins? Is it always the same?
Can you do it by using a loop in stead?

Optional exercise

Create an Alice world that shows a search and rescue operation in which a helicopter flies in a pattern above the ocean, occasionally stopping to hover for a while as if looking for something.
To do that create first a method called “hover” for the helicopter that accepts the number of seconds it should hover as a parameter. Create a search method and a rescue method to handle these aspects of the operation.
The world may look as below: