29.10.2015

Nils Gawlik
(Other group members: Jannes Brunner, Oliver Christ)

Lab Report for the 27th October 2015

Note:

I wrote the pre-lab solutions in some sort of pseudo-code: Indentation structures the code into blocks, method definitions are indicated by colons, method calls do not require brackets and "stop" stands for "Greenfoot.stop(); return;".

Exercise 1:

```
Pre-Lab solution:
set counter to 0
act:
  if counter < 10
    putLeaf
    move
  else
    stop
  increment counter by 1
```

The direct implementation of the pseudo-code worked, the other group members had a very similar approach. Oliver likes to use an extra for-loop, instead of the run button.

Exercise 2:

```
Pre-Lab solution:
set inTunnel to false
act:
  if treeRight and treeLeft
    putLeaf
    set inTunnel to true
  else
    if inTunnel //leaving the tunnel
      stop
  move
```

There were no problems implementing the code. Jannes tried to get his version working, using an xor-operator to detect the trees and I helped him with some small things and in the end that worked, too.

29.10.2015

Nils Gawlik
(Other group members: Jannes Brunner, Oliver Christ)

Exercise 3:

```
Pre-Lab solution:
act:
  if treeFront
    avoidTree
  else
    move

avoidTree:
  turnLeft move turnRight //sidestep
  while treeRight
    move
  turnRight move turnLeft //back on track
```

The pre-lab solution is in probably correct in principle, but the leaf detection is missing; there also needs to be another move before the while loop, to avoid the same mistake of going back down instantly I encountered later in this lab.

I also wanted to make a version which didn't rely on one more nested loop (counting the run button as loop already). So I wrote a completely different algorithm which relies on a "top" variable. This variable represents if Kara is in the top row currently avoiding the trees. Kara now moves forward every step and avoids trees if in the bottom row and tries to go back down if in the top row.

This caused a problem where Kara would go to the top row and then instantly to the bottom row again, so I added an extra move forward. This has the side effect that Kara can't stop on a leaf immediately after entering the top row. Kara can stop on any other field though, which is nice.

Exercise 4:

```
Pre-Lab solution:
act:
  if treeFront
    if not treeRight
      turnRight
    else
      turnLeft //no further checking necessary if forest is built correctly
  move
```

I forgot to check for the leaf – again. I also changed the if condition to treeLeft() because it seemed cleaner.

29.10.2015

Nils Gawlik
(Other group members: Jannes Brunner, Oliver Christ)

Exercise 5:

Pre-Lab solution:

```
set facingEast to true
act:
  //inversion part:
  if onLeaf
    removeLeaf
  else
    putLeaf

  //movement part:
  if not treeFront
    move
  else   //turning around at wall
    if facingEast
      if treeRight
        stop
      turnRight
      move
      turnRight
    else
      if treeLeft
        stop
      turnLeft
      move
      turnLeft
```

I forgot to set the variable after turning, other than that, the algorithm was fine.

Exercise 6:

Pre-Lab solution:

```
act:
  if length < 20
    putTrail(length)
    increase length by 1
    turnRight
  else
    stop

putTrail(n):
  repeat n times
    putLeaf
    move
```

The algorithm worked. I just shortened some things.