Abraham Gebrekidan
EIP 01

Reading #4A
July 26, 2024

Solutions to the Banish Winter Problem

```
/*
 * File: BanishWinter.k
 * --------------------
 * The BanishWinter program gets Karel adorn a series of trees with a
 * cluster of beeper leaves.
 */

import "turns";

/* Main function */

function banishWinter() {
   while (beepersInBag()) {
      findTree();
      addLeavesToTree();
   }
}

/*
 * Moves Karel up to the next tree.
 *
 * Programming style note: Since a tree is simply a wall, this method
 * can simply call moveToWall.  You could therefore replace the
 * findTree call in the main program with moveToWall, but the program
 * might then be harder to read because it violates the "tree" metaphor
 * used at the level of the main program.
 */

function findTree() {
   moveToWall();
}

/*
 * Adorns a single tree with a cluster of leaves.  The precondition
 * is that Karel must be immediately west of the tree, facing east;
 * the postcondition is that Karel is at the bottom of the other side
 * of the tree, facing east.
 */

function addLeavesToTree() {
   turnLeft();
   climbTree();
   addLeaves();
   descendToGround();
   turnLeft();
}
```

```
/*
 * Climbs to the top of the tree.
 */

function climbTree() {
   while (rightIsBlocked()) {
      move();
   }
}

/*
 * Moves Karel back to the ground.
 */

function descendToGround() {
   moveToWall();
}

/*
 * Creates the cluster of leaves at the top of a tree.  The
 * precondition is that Karel must be facing north at the top
 * of the tree; the postcondition is that Karel is still at the
 * top, but on the other side of the trunk, facing south.
 */

function addLeaves() {
   turnRight();
   makeBeeperSquare();
   move();
   turnRight();
}

/*
 * Moves Karel forward until it is blocked by a wall.
 */

function moveToWall() {
   while (frontIsClear()) {
      move();
   }
}

/*
 * Creates a square of four beepers, leaving Karel in its original
 * orientation.  The resulting square is positioned ahead and to the
 * left of Karel's starting position.
 */

function makeBeeperSquare() {
   repeat (4) {
      putBeeper();
      move();
      turnLeft();
   }
}
```