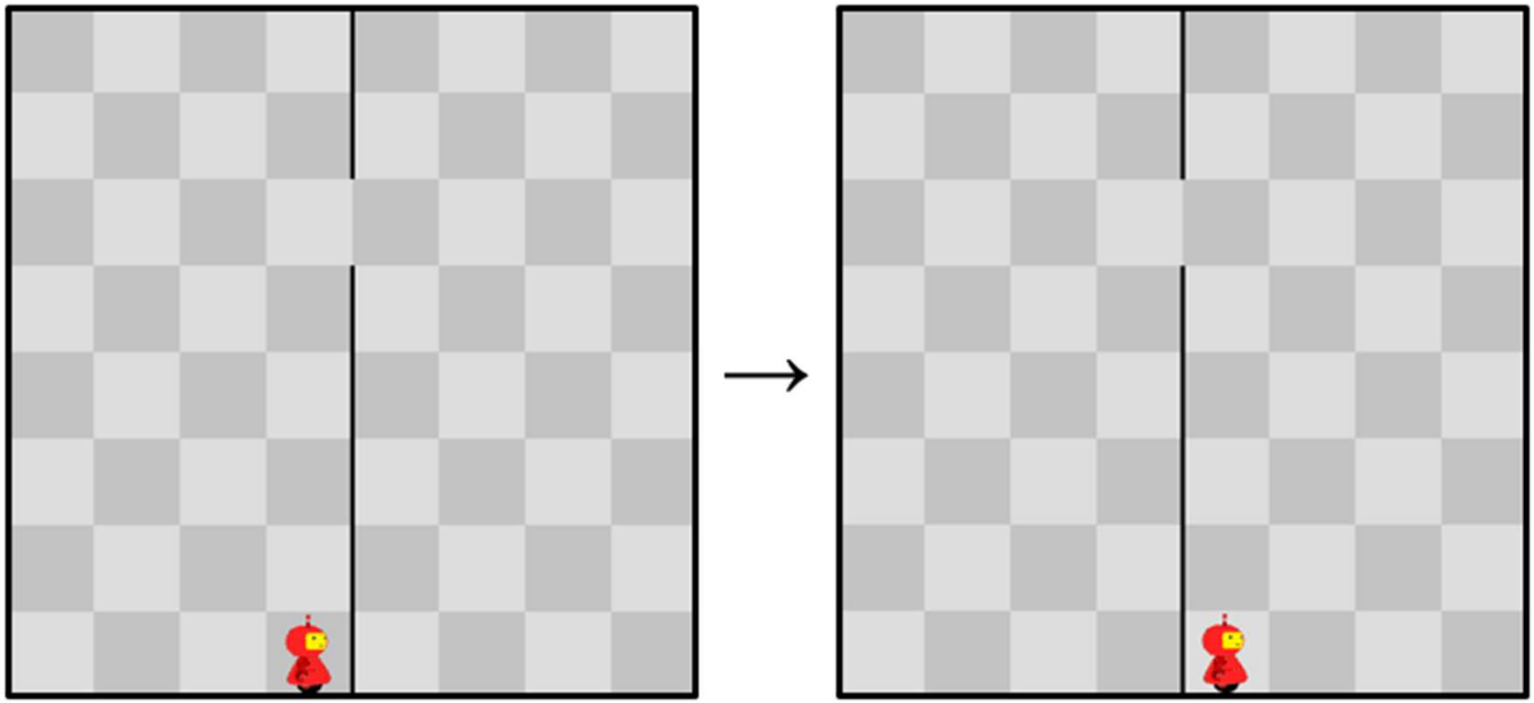

PA 1
Karel Explores his World

PA 1

wall.c

DISCUSSION



```
BYPASS_WALL
```

```
GO_UP_TO_DOOR
MOVE
GO_ALL_THE_WAY_DOWN
```

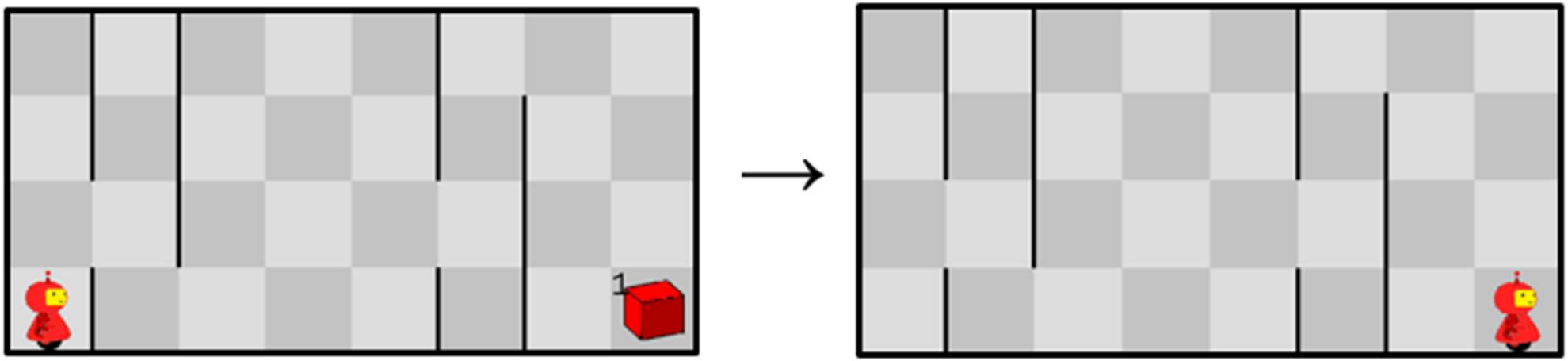
DISCUSSION

```
// He starts facing E. Then moves N along the wall
// until there is an opening. He ends up facing E.
void go_up_to_door() {
    turn_left();
    while (wall_to_right())
          move();
    turn_right();
}

// He starts facing E. Then moves S along the wall until he hits a wall.
// He ends up facing E.
void go_all_the_way_down() {
    turn_right();
    while (!wall_in_front())
          move();
    turn_left();
}

// This function assumes Karel is facing E against a wall. It will have him climb up
// until he finds an opening, move through it (in the E direction) and then climb all
// the way the down on the other end, and turn so he is facing E again.
void bypass_the_wall() {
    go_up_to_door();
    move();
    go_all_the_way_down();
}

int main() {
    karel_setup("settings/settings01_wall.json");
    bypass_the_wall();
    turn_off();
}
```

PA 1

doors.c

DISCUSSION



```
GO_TO_WALL
WHILE (NOT FOUND ITEM) {
        BYPASS_WALL
        GO_TO_WALL
}
```

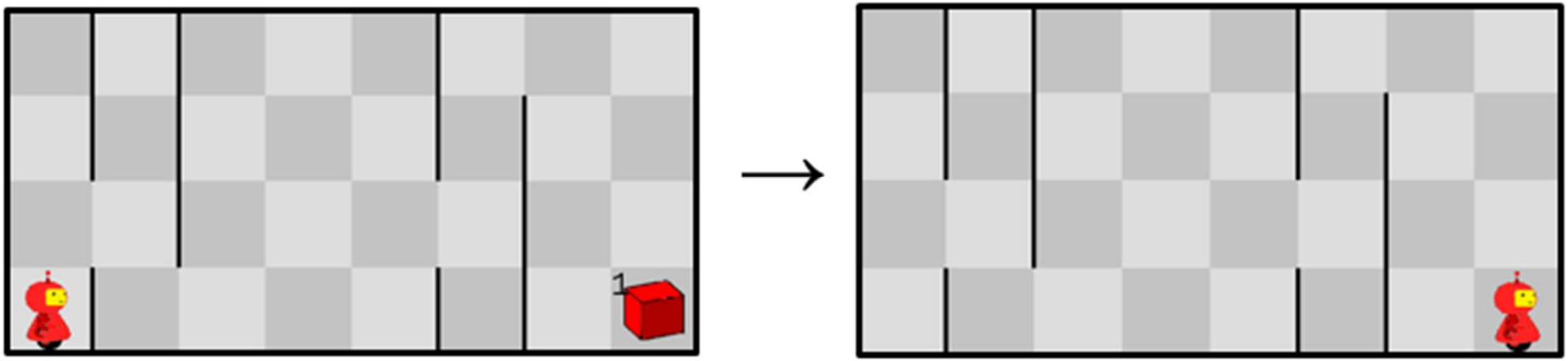
```
void go_to_wall() {
    while (!wall_in_front())
          move();
}

void go_up_to_door() {
    turn_left();
    while (wall_to_right())
          move();
    turn_right();
}

void go_all_the_way_down() {
    turn_right();
    while (!wall_in_front())
          move();
    turn_left();
}

void bypass_the_wall() {
    go_up_to_door();
    move();
    go_all_the_way_down();
}

int main() {
    karel_setup("settings/settings01_doors.json");
    go_to_wall();
    while (!item_present()) {
          bypass_the_wall();
          go_to_wall();
    }
    take_item();
    turn_off();
}
```

DISCUSSION

PA 1

doors.c

DISCUSSION

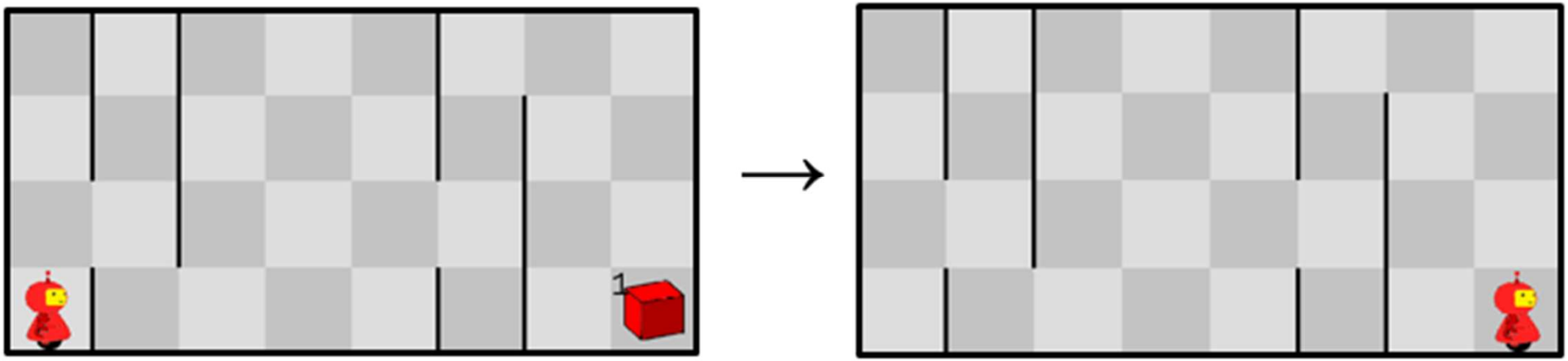



```
FOREVER {
        GO_TO_WALL
        IF (ITEM FOUND)
                STOP
        BYPASS_WALL


}
```

PA 1

doors.c

DISCUSSION

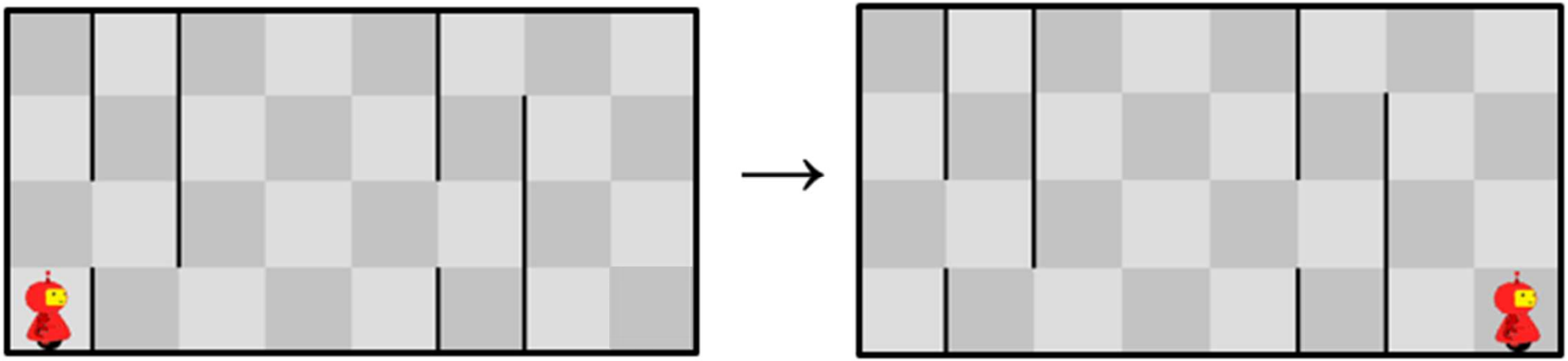



```
FOREVER {
        GO_TO_WALL
        IF (ITEM FOUND)
                STOP
        GO_UP_TO_DOOR
        MOVE
        GO_ALL_THE_WAY_DOWN
}
```

PA 1

missing.c

DISCUSSION



```
FOREVER {
        GO_TO_WALL
        FIND_DOOR_OR_CORNER
        IF (CORNER)
                GO_DOWN_AND_STOP
        ELSE
                MOVE
                GO_ALL_THE_WAY_DOWN
}
```

```
void go_to_wall() {
    while (!wall_in_front())
          move();
}
```

DISCUSSION

```
void go_up_to_door_or_wall() {
    turn_left();
    while ( wall_to_right() && !wall_in_front() )
          move();
    turn_right();
}
```

Stop when (i.e., go until) there is
no wall to the right or a wall in front

```
void go_all_the_way_down() {
    turn_right();
    go_to_wall();
    turn_left();
}
```

```
while ( ! STOP )
          move();
```

```
while ( ! (!wall_to_right() || wall_in_front()) )
          move();
```

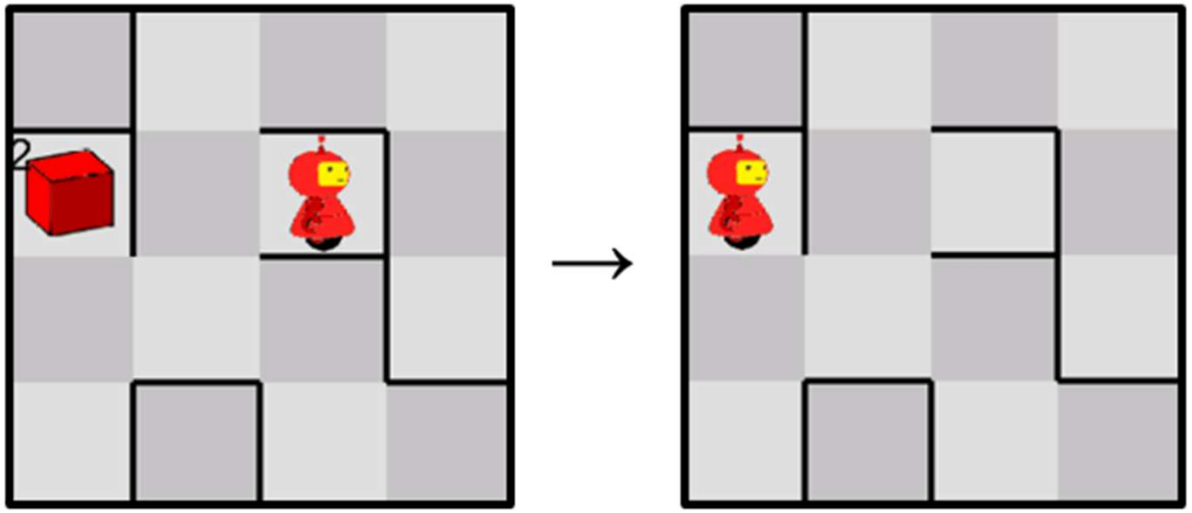
```
int main() {
    karel_setup("settings/settings01_missing.json");
    while (!item_present() || item_present()) {
          go_to_wall();
          go_up_to_door_or_wall();
          if (wall_in_front()) {
              go_all_the_way_down();
              turn_off();
          }
          else {
              move();
              go_all_the_way_down();
          }
    }
    turn_off();
}
```

Always true
(trick to make infinite loop)

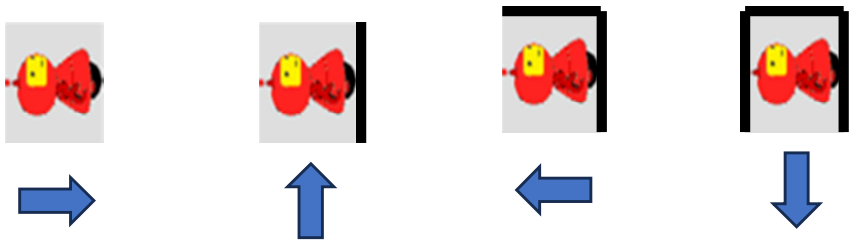
PA 1

maze.c

DISCUSSION



```
WHILE (NO ITEM) {
        CHOOSE_DIRECTION
        MOVE
}
PICK_UP_TWO_ITEMS
```



```
#include <karel.h>

void choose_direction() {
    if (!wall_to_right())
          turn_right();
    else
        if (!wall_in_front())
          ;
        else
            if (!wall_to_left())
                turn_left();
            else {
                turn_left();
                turn_left();
            }
}

void take_two_items() {
    take_item();
    take_item();
}

// This is where the C-program starts
int main() {
    karel_setup("settings/settings01_maze.json");

    while (!item_present()) {
          choose_direction();
          move();
    }
    take_two_items();

    turn_off();
}
```

DISCUSSION

Empty statement (is needed here)

If-Else Chain

DISCUSSION

```
if ( query )
        statement
else
        statement
```

```
void choose_direction () {

   if (!wall_to_right())
         turn_right();
   else
       if (!wall_in_front())
             ;
       else
           if (!wall_to_left())
                 turn_left();
           else {
                 turn_left();
                 turn_left();
            }
}
```

```
void choose_direction () {
   if (!wall_to_right())
         turn_right();
   else if (!wall_in_front())
         ;
   else if (!wall_to_left())
         turn_left();
   else {
         turn_left();
         turn_left();
   }
}
```

```
#include <karel.h>

void choose_direction() {
    if (!wall_to_right())
          turn_right();
    else if (!wall_in_front())
          ;
    else if (!wall_to_left())
          turn_left();
    else {
          turn_left();
          turn_left();
    }
}

void take_two_items() {
    take_item();
    take_item();
}

// This is where the C-program starts
int main() {
    karel_setup("settings/settings01_maze.json");

    while (!item_present()) {
          choose_direction();
          move();
    }
    take_two_items();

    turn_off();

}
```

DISCUSSION

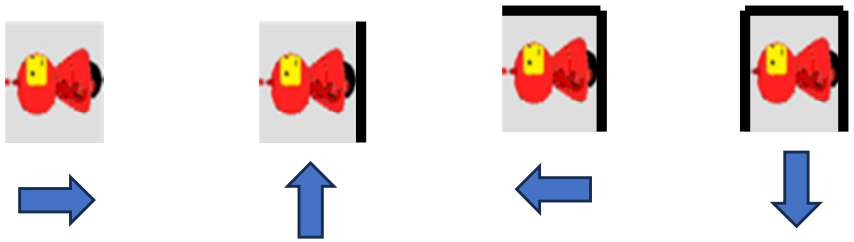
```
#include <karel.h>

void choose_direction_and_move() {
    if (!wall_to_right()) {
        turn_right();
        move();
    }
    else if (!wall_in_front())
        move;
    else if (!wall_to_left()) {
        turn_left();
        move();
    }
    else {
        turn_left();
        turn_left();
        move();
    }
}

void take_two_items() {
    take_item();
    take_item();
}

// This is where the C-program starts
int main() {
    karel_setup("settings/settings01_maze.json");

    while (!item_present())
        choose_direction_and_move();
    take_two_items();

    turn_off();
}
```

DISCUSSION

This is an alternative way of structuring the solution …

If-Else or If

DISCUSSION



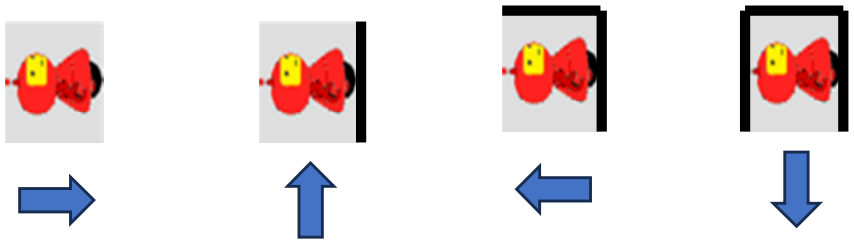
```
void choose_direction_and_move() {
     if (!wall_to_right()) {
          turn_right();
          move();
     }
     else if (!wall_in_front())
          move();
     else if (!wall_to_left()) {
          turn_left();
          move();
     }
     else {
          turn_left();
          turn_left();
          move();
     }
}
```

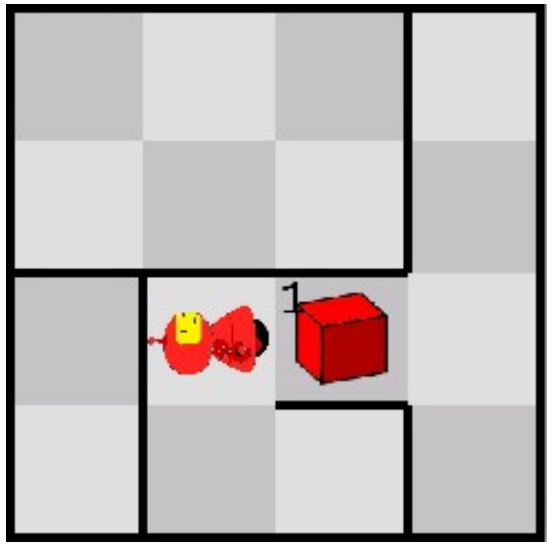
Does this work as well?

```
void choose_direction_and_move() {
     if (!wall_to_right()) {
          turn_right();
          move();
     }
     if (!wall_in_front())
          move();
     if (!wall_to_left()) {
          turn_left();
          move();
     }
     else {
          turn_left();
          turn_left();
          move();
     }
}
```

If-Else or If

DISCUSSION





```
int main() {
    karel_setup("settings/settings01_maze.json");
    while (!item_present())
          choose_direction_and_move();
    take_two_items();
    turn_off();
}
```

Does this work as well? No!

```
void choose_direction_and_move() {
    if (!wall_to_right()) {
          turn_right();
          move();
    }
    if (!wall_in_front())
          move();
    if (!wall_to_left()) {
          turn_left();
          move();
    }
    else {
          turn_left();
          turn_left();
          move();
    }
}
```

If and While

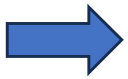
DISCUSSION

Some other points that often come up …

```
while (!item_present())
    if (!item_present())
        choose_direction_and_move();
```

```
while (!item_present())
    choose_direction_and_move();
```

```
if (!wall_to_left())
    turn_left();
else if (wall_to_left()) {
    turn_left();
    turn_left();
}
```



```
if (!wall_to_left())
    turn_left();
else {
    turn_left();
    turn_left();
}
```