

# [ai] explore in Logo 2 (for advanced programming)

## Prior knowledge

The expected prior knowledge in the field of programming and computational thinking was:

- creates an algorithm to solve a simple problem, checks the correctness of the algorithm, detects and corrects errors,
- uses a programming tool to create a program that uses input and output values and repetition.

Suggested tasks for testing prior knowledge (at the end of the document)

Learning outcomes:

- creates, monitors, and refactors programs that contain branching and conditional repetition structures and predicts the behaviour of simple algorithms that can be represented by diagrams, spoken words, or programming language,
- considers and solves a more complex problem by breaking it down into a series of subproblems.

## Preparation for the activity

- The game [ai] explore! (Appendix 1)
- 5x5 Geoboards or paper with prepare boards (ideally one per student/computer). In the following activity, we used the first game in the Appendix 1
- Rubber bands or pencils

## WEEK 1 (2 x 45min)

1. Playing the [ai] explore! game
2. Construction of polygons in FMSLogo (excercise) using repeat for drawing the square

**Task 1.** Make a procedure “square” which draws a square with side of length 50 and color the square with a color of your choice.

### Solution 1

*to square*

*repeat 4 [fd 50 rt 90]*

*pu fd 25 rt 90 fd 25 setfc 5 fill*

*end*

**Task 2.** Modify the procedure "square" from Task 1 so that it draws a square whose side is of length  $d$  (the user enters the side length outside the procedure).

### Solution 2

```
to square :d
  repeat 4 [fd :d rt 90]
  pu fd :d/2 rt 90 fd :d/2 setfc 5 fill pd
end
```

```
make "d readword
square :d
```

**Task 3.** Modify the procedure "square" from Task 2 so that it draws a square whose side is of length  $d$  and colors it with color  $b$  which user enters (inside of the procedure).

After the square is drawn, set Turtle to lower left corner of the square, looking upwards (starting position).

**COMMANDS:** make, readword, repeat, setfc :color fill, invoking a procedure with a variable; color code

### Solution 3

```
to square :d
  repeat 4 [fd :d rt 90]
  print "color?"
  make "color readword
  pu fd :d/2 rt 90 fd :d/2 setfc :color fill
  bk :d/2 lt 90 bk :d/2 pd
end
```

```
print "d=?"
make "d readword
square :d
```

**Task 4.** Using the procedure created in Task 3, make 5x5 square grid such that each square is colored (as in the geoboard in front of you). Hint: make a plan how to draw a 5x5 grid (for example, row by row or column by column).

**COMMANDS :** for

**Solution 4**

*// PLAN: draw line by line (from left to right) by drawing the top line first and then moving down. The turtle ends under the last line, under the left square, as in the Figure 1.*

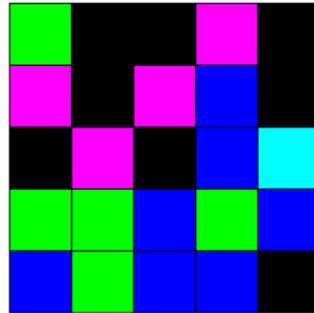


Figure 1.

```
to square :d
  repeat 4 [fd :d rt 90]
  print "color?"
  make "color readword
  pu fd :d/2 rt 90 fd :d/2 setfc :color fill
  bk :d/2 lt 90 bk :d/2 pd
end
```

```
print "d=?"
make "d readword
```

```
for [i 1 5 1][
  square :d
  pu rt 90 fd :d lt 90 pd
]
pu lt 90 fd :d*5 lt 90 fd :d lt 180 pd
```

```
for [i 1 5 1][
  square :d
  pu rt 90 fd :d lt 90 pd
]
pu lt 90 fd :d*5 lt 90 fd :d lt 180 pd
```

```

for [i 1 5 1][
  square :d
  pu rt 90 fd :d lt 90 pd
]
pu lt 90 fd :d*5 lt 90 fd :d lt 180 pd
for [i 1 5 1][
  square :d
  pu rt 90 fd :d lt 90 pd
]
pu lt 90 fd :d*5 lt 90 fd :d lt 180 pd
for [i 1 5 1][
  square :d
  pu rt 90 fd :d lt 90 pd
]
pu lt 90 fd :d*5 lt 90 fd :d lt 180 pd

```

## WEEK 2 (2x 45min)

At the beginning of the exercise, all students have access to the file with solution of Task 4 (from week 1).

**Task 5.** Place the turtle in the middle of the top left square (from Task 4) and orient it so that it faces right. Walk the turtle across the grid of squares so that it follows the same path as indicated on the Geoboard in front of you.

### Solution 5

```

to square :d
  repeat 4 [fd :d rt 90]
  print "color?"
  make "color readword
  pu fd :d/2 rt 90 fd :d/2 setfc :color fill
  bk :d/2 lt 90 bk :d/2 pd
end

print "d=?"
make "d readword
for [i 1 5 1][
  square :d
  pu rt 90 fd :d lt 90 pd
]
pu lt 90 fd :d*5 lt 90 fd :d lt 180 pd

```

```

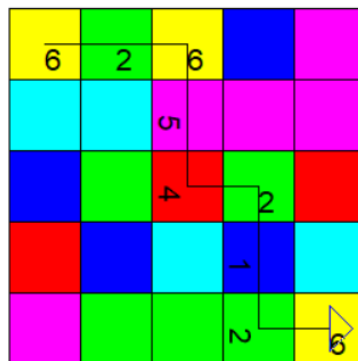
for [i 1 5 1][
  square :d
  pu rt 90 fd :d lt 90 pd
]
pu lt 90 fd :d*5 lt 90 fd :d lt 180 pd
for [i 1 5 1][
  square :d
  pu rt 90 fd :d lt 90 pd
]
pu lt 90 fd :d*5 lt 90 fd :d lt 180 pd
for [i 1 5 1][
  square :d
  pu rt 90 fd :d lt 90 pd
]
pu lt 90 fd :d*5 lt 90 fd :d lt 180 pd
for [i 1 5 1][
  square :d
  pu rt 90 fd :d lt 90 pd
]
pu lt 90 fd :d*5 lt 90 fd :d lt 180 pd
pu fd :d*5 fd :d/2 rt 90 fd :d/2 pd
fd :d fd :d rt 90 fd :d fd :d lt 90 fd :d rt 90 fd :d fd :d lt 90 fd :d

```

**Task 6.** Modify the Task 5 such that turtle writes the color code of that square on each square on its path.

**COMMANDS:** label pixel

Solution 6



```

to square :d
  repeat 4 [fd :d rt 90]
  print "color?"
  make "color readword
  pu fd :d/2 rt 90 fd :d/2 setfc :color fill
  bk :d/2 lt 90 bk :d/2 pd
end

print "d=?"
make "d readword
for [i 1 5 1][
  square :d
  pu rt 90 fd :d lt 90 pd
]
pu lt 90 fd :d*5 lt 90 fd :d lt 180 pd
for [i 1 5 1][
  square :d
  pu rt 90 fd :d lt 90 pd
]
pu lt 90 fd :d*5 lt 90 fd :d lt 180 pd
for [i 1 5 1][
  square :d
  pu rt 90 fd :d lt 90 pd
]
pu lt 90 fd :d*5 lt 90 fd :d lt 180 pd
for [i 1 5 1][
  square :d
  pu rt 90 fd :d lt 90 pd
]
pu lt 90 fd :d*5 lt 90 fd :d lt 180 pd
for [i 1 5 1][
  square :d
  pu rt 90 fd :d lt 90 pd
]
pu lt 90 fd :d*5 lt 90 fd :d lt 180 pd
pu fd :d*5 fd :d/2 rt 90 fd :d/2 pd

label pixel
fd :d label pixel
fd :d label pixel
rt 90 fd :d label pixel

```

```
fd :d label pixel
lt 90 fd :d label pixel
rt 90 fd :d label pixel
fd :d label pixel
lt 90 fd :d label pixel
```

## WEEK 3 (2x 45min)

**Task 7.** Walk the turtle around the grid so that it starts from the upper left corner and moves only to the right or down using the buttons along the path indicated on the Geoboard.

**COMMANDS:** buttons

### Solution 7

```
to square :d
  repeat 4 [fd :d rt 90]
  print "color?"
  make "color readword
  pu fd :d/2 rt 90 fd :d/2 setfc :color fill
  bk :d/2 lt 90 bk :d/2 pd
end
```

```
BUTTONCREATE "main "myright "DOWN -100 100 50 25 [pd rt 90 fd :d lt 90 label pixel]
BUTTONCREATE "main "mydown "RIGHT -50 100 50 25 [pd fd :d label pixel]
```

```
print "d=?"
make "d readword
```

```
for [i 1 5 1][
  square :d
  pu rt 90 fd :d lt 90 pd
]
pu lt 90 fd :d*5 lt 90 fd :d lt 180 pd
for [i 1 5 1][
  square :d
  pu rt 90 fd :d lt 90 pd
]
pu lt 90 fd :d*5 lt 90 fd :d lt 180 pd
```

```

for [i 1 5 1][
  square :d
  pu rt 90 fd :d lt 90 pd
]
pu lt 90 fd :d*5 lt 90 fd :d lt 180 pd
for [i 1 5 1][
  square :d
  pu rt 90 fd :d lt 90 pd
]
pu lt 90 fd :d*5 lt 90 fd :d lt 180 pd
for [i 1 5 1][
  square :d
  pu rt 90 fd :d lt 90 pd
]
pu lt 90 fd :d*5 lt 90 fd :d lt 180 pd

pu fd :d*5 fd :d/2 rt 90 fd :d/2 pd
label pixel

```

**before restarting the program:**

```

buttondelete "myright
buttondelete "mydown

```

Creating the plan to make a game on 8x8 grid. (45 min)

1. Show the student a game in Logo, i.e. the final goal for students to create

**Task 8.** Enter the grid size  $n$  ( $n=8$ ) and make an  $n \times n$  grid such that the squares are colored with random color (some of the five colors: black, blue, green, yellow, red).

**COMMANDS:** pick [0 1 2 4 6]

### Solution 8

```

to square :d
  repeat 4 [fd :d rt 90]
  make "color pick [0 1 2 4 6]
  pu fd :d/2 rt 90 fd :d/2 setfc :color fill
  bk :d/2 lt 90 bk :d/2 pd
end

```

```

print "n=?"
make "n readword

print "d=?"
make "d readword

BUTTONCREATE "main "myright "DOWN -100 100 50 25 [pd rt 90 fd :d lt 90 label pixel]
BUTTONCREATE "main "mydown "RIGHT -50 100 50 25 [fd :d label pixel]

for [j 1 :n 1][
  for [i 1 :n 1][
    square :d
    pu rt 90 fd :d lt 90 pd
  ]
  pu lt 90 fd :d*n lt 90 fd :d lt 180 pd
]
pu fd :d*n fd :d/2 rt 90 fd :d/2 pd
label pixel

before restarting the program:
buttondelete "myright
buttondelete "mydown

```

## WEEK 4 (2x45min)

**Task 9.** Make a procedure “points” which reads the color of the field where the turtle is and writes the number of points that color carries, respecting the following scale: black - 1 point, blue - 2 points, green - 3 points, yellow - 4 points, red - 5 points).

Modify task 8 so that the turtle writes the number of points in each field (instead of the field color label).

### Solution 9

```

to points
  if pixel=0 [label 1]
  if pixel=1 [label 2]
  if pixel=2 [label 3]
  if pixel=4 [label 5]
  if pixel=6 [label 4]
end

to square :d
  repeat 4 [fd :d rt 90]
  make "color pick [0 1 2 4 6]

```

```

    pu fd :d/2 rt 90 fd :d/2 setfc :color fill
    bk :d/2 lt 90 bk :d/2 pd
end

print "n=?"
make "n readword

print "d=?"
make "d readword

BUTTONCREATE "main "myright "DOWN -100 100 50 25 [pd rt 90 fd :d lt 90 points]
BUTTONCREATE "main "mydown "RIGHT -50 100 50 25 [fd :d points]

for [j 1 :n 1][
    for [i 1 :n 1][
        square :d
        pu rt 90 fd :d lt 90 pd
    ]
    pu lt 90 fd :d*:n lt 90 fd :d lt 180 pd
]
pu fd :d*:n fd :d/2 rt 90 fd :d/2 pd
points

before restarting the program:
buttondelete "myright
buttondelete "mydown

```

## WEEK 5 (2x45min)

**Task 10.** Modify task 9 so that the turtle walks on the board from the upper left corner and moves only to the right or down using the buttons and, using the "points" procedure, calculates the points it has collected during its walk.

**COMMAND:** if

### Solution 10

```

to points
    if pixel=0 [make "score :score + 1]
    if pixel=1 [make "score :score + 2]
    if pixel=2 [make "score :score + 3]
    if pixel=4 [make "score :score + 5]
    if pixel=6 [make "score :score + 4]
    print :score
end

```

```

to square :d
  repeat 4 [fd :d rt 90]
  make "color pick [0 1 2 4 6]
  pu fd :d/2 rt 90 fd :d/2 setfc :color fill
  bk :d/2 lt 90 bk :d/2 pd
end

```

```

make "score 0
print "n=?
make "n readword

print "d=?
make "d readword

```

```

BUTTONCREATE "main "myright "DOWN -100 100 50 25 [pd rt 90 fd :d lt 90 points]
BUTTONCREATE "main "mydown "RIGHT -50 100 50 25 [fd :d points]

```

```

for [j 1 :n 1][
  for [i 1 :n 1][
    square :d
    pu rt 90 fd :d lt 90 pd
  ]
  pu lt 90 fd :d*n lt 90 fd :d lt 180 pd
]
pu fd :d*n fd :d/2 rt 90 fd :d/2 pd
points

```

**before restarting the program:**

```

buttondelete "myright
buttondelete "mydown

```

## WEEK 6 (2x45min)

**Task 11.** Modify the task 10 in a way that turtle cannot walk outside of the grid and when the turtle comes to the lower right corner write down the total amount of points.

**COMMANDS:** if – else (if number of steps to the right equals  $n$  then ALERT, else step++)

## Solution 11

to points

```

    if pixel=0 [make "score :score + 1]
    if pixel=1 [make "score :score + 2]
    if pixel=2 [make "score :score + 3]
    if pixel=4 [make "score :score + 5]
    if pixel=6 [make "score :score + 4]

```

end

to square :d

```

    repeat 4 [fd :d rt 90]
    make "color pick [0 1 2 4 6]
    pu fd :d/2 rt 90 fd :d/2 setfc :color fill
    bk :d/2 lt 90 bk :d/2 pd

```

end

make "score 0

print "n=?"

make "n readword

print "d=?"

make "d readword

make "stepsRight 0

make "stepsDown 0

```

BUTTONCREATE "main "myright "DOWN -100 100 50 25 [ifelse :stepsDown<:n-1 [pd rt
90 fd :d lt 90 points make "stepsDown :stepsDown + 1] [print "No_more_down]
if :stepsRight+:stepsDown=2*:n-2 [label :score]]

```

```

BUTTONCREATE "main "mydown "RIGHT -50 100 50 25 [ifelse :stepsRight<:n-1 [pd fd :d
points make "stepsRight :stepsRight + 1] [print "No_more_right]
if :stepsRight+:stepsDown=2*:n-2 [label :score]]

```

for [j 1 :n 1][

for [i 1 :n 1][

square :d

pu rt 90 fd :d lt 90 pd

]

pu lt 90 fd :d\*:n lt 90 fd :d lt 180 pd

```
]
pu fd :d*:n fd :d/2 rt 90 fd :d/2 pd
points
```

**Task 12.** Testing the created program.

## WEEK 7 (2x45min)

### PROJECT TASK:

Enter the size of the grid  $n$  ( $n=8$ ) and draw a square grid  $n \times n$  so that the squares are colored randomly (some color from the 5 colors: black, blue, green, yellow, red), but in such a way that the first column is colored as follows: blue, green, red, yellow, red (so that the blue square is in the upper left corner). If there are more than 5 rows, a random color is chosen for the rest of the first column. Finally, place the turtle in the lower right square and orient it to face right.

Walk the turtle around the grid so that it starts from the lower right corner (and ends in the upper left corner) and moves only left or up using the buttons (after each step the turtle looks to the right again).

Create a procedure "points" that reads the color of the square the turtle is on and returns the number of points that that color carries, respecting the following scale: black - 0 points, blue - 1 point, green - 2 points, yellow - 3 points, red - 4 points. The turtle writes the number of points it has collected so far in each space it lands on.

### Solution of project task

```
to points
  if pixel=1 [make "score :score + 1]
  if pixel=2 [make "score :score + 2]
  if pixel=4 [make "score :score + 4]
  if pixel=6 [make "score :score + 3]
  label :score
end
```

```
to square :d
  repeat 4 [fd :d rt 90]
  make "color pick [0 1 2 4 6]
```

```

    if :i=:j [make "color 2]
    pu fd :d/2 rt 90 fd :d/2 setfc :color fill
    bk :d/2 lt 90 bk :d/2 pd
end

make "score 0
print "n=?"
make "n readword

print "d=?"
make "d readword

make "stepsRight 0
make "stepsDown 0

BUTTONCREATE "main "myright "down -120 100 50 25 [ifelse :stepsDown<:n-1 [pd rt 90
fd :d lt 90 points make "stepsDown :stepsDown +1] [print "No_more_down]
if :stepsRight+:stepsDown=2*:n-2 [print :score]]

BUTTONCREATE "main "mydown "DESNO -70 100 50 25 [ifelse :stepsRight<:n-1 [pd fd :d
points make "stepsRight :stepsRight +1] [print "No_more_right]
if :stepsRight+:stepsDown=2*:n-2 [print :score]]

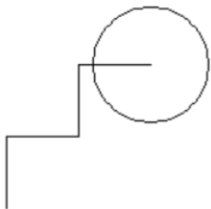
for [j 1 :n 1][
  for [i 1 :n 1][
    square :d
    pu bk :d pd
  ]
  pu fd :d*n rt 90 fd :d lt 90 pd
]
pu bk :d*(n-1) lt 90 fd :d/2 rt 90 fd :d/2 rt 90 pd
points

```

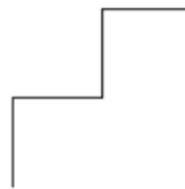
## Suggested tasks for testing prior knowledge

- \_\_\_ is the command that starts the program; while the command \_\_\_ is the command that ends it.
- Which command do we use to fill a shape?
  - fillshape
  - fill
  - repeat
  - pu
- Which commands will draw a rectangle (2 correct answers)?
  - repeat 4 [fd 60 rt 90]*
  - repeat 2 [fd 60 rt 90 fd 40 rt 90]*
  - repeat 2 [fd 60 rt 90 fd 60 rt 90]*
  - fd 60 rt 90 fd 40 rt 90 fd 60 rt 90 fd 40 rt 90*
- What shape will we get with command: *repeat 3 [fd 75 lt 360/3]*?
  - rectangle
  - square
  - triangle
  - circle
- Which drawing will we get with the following commands:

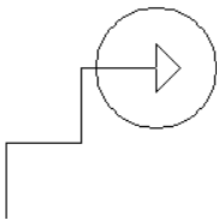
```
repeat 2 [fd 50 rt 90]
lt 180 fd 50
rt 90 fd 50
circle 40
ht
```



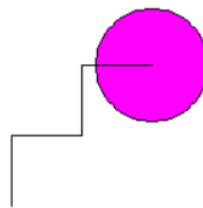
a)



b)



c)



d)

6. Write the commands that will make the turtle go back 150 steps and rotate to the left by 45°.
  
  
  
  
  
  
  
  
  
  
7. Write the commands you will use to draw the letter H.
  
  
  
  
  
  
  
  
  
  
8. Change the program to use variables. The variable :n represents the number of sides and the number of angles of the figure, and the variable :d represents the length of the side of the figure.

```
to shape  
  repeat 7 [fd 80 lt 360/7]  
end
```