Appendix 1: assignment booklet challenge 9 - 22

Kojo and chess

Challenge 9:

Draw a chess board with the command:

```
board
```

Add notation with the command:

```
notation
```

The turtle has now drawn a chess board with coordinates.

Challenge 10:

Choose different starting points for the turtle in the first row with the command:

```
startingPoint("A1")
```

The code gives us the starting point A1 for the turtle. Start with choosing different starting points in the first row: A1,B1, … H1, after that you can choose any starting point on the chessboard.

Challenge 11:
Write a code for the turtle to move from D1 to B7, test different ways for the turtle to move.

Choose different starting points in the first row and choose a finish point on the board.
Try with the command `repeat( ) {   }`

Movements of other chess pieces:

We can move the **bishop** with the commands "`diagonalForwardRight`", "`diagonalForwardLeft`", "`diagonalBackwardRight`", "`diagonalBackwardLeft`"

Challenge 12:
Write a code for the turtle to move the **bishop** from D1 to H7

Use the command `repeat( ) {   }` to simplify your code. In order to follow the turtles path, use the command:

```
setSpeed(slow)
```

Challenge 13:

What commands are you allowed to use for the queen? Write a code that moves the **queen** from C1 to G8

Make your own function with def

for some guidance: **film**: make a definition in kojo

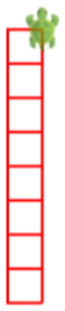
Make a def in kojo: https://youtu.be/D0UzBldOhWI

Challenge 14:

Use the code:

```
def square = repeat(4) { forward;  left }
square
```

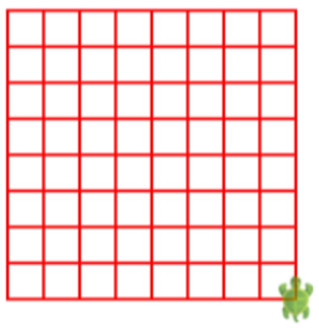
//Tip: you need to write square if you want the turtle to draw a square!



Challenge 15: build a stack with 8 squares

Use: def square =… and the command hop

Challenge 16: make a 8x8 grid. Tip: Use a stack function and a the command hop (-200)



the following challenges are optional:

Challenge 17: draw a stair with a lot of steps, make a function ”stair”.

Add comments in your code

```
//
```

The program doesn’t run anything that follows //, what follows turns into comments.

Add some comments to your last code.

You can even deactivate parts of the code with //

If you want to mark a larger part, you do this with /* and */

Draw larger or smaller squares

Challenge 18: test the following code:

```
def square100= repeat( 4 ) { forward(100) ; right}

def square200= repeat( 4 ) { forward(200) ; right}



def square50= repeat( 4 ) { forward(50) ; right}

def square20= repeat( 4 ) { forward(20) ; right}
```

Square with parameters

film: make a def with a parameter

```
def  square( length:Int ) = repeat( 4 ) { forward(length) ; right}
```

now you can draw different sized squares by writing, for example:

```
square(100)
```

Challenge 19: draw different sized squares with your function above.

Challenge 20: change the color of your squares with the command color. There are pre-defined colors such as green, blue, red, yellow…

Challenge 21: Make a function for a stair with step height and length as parameters.

for some guidance how to use parameters:

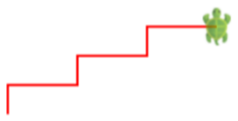
film:

tip: use

```
def step( length:Int , height:Int ) =
{ forward(length) ; left; forward(height) ; right}
```

Now you can draw different sized steps by writing, for example:

```
step( 100, 20 )
```



Challenge 22 : draw a rectangle with width and length as parameters.

Tip:

```
def rectangle ( h:Int  , b:Int ) = { …
```

now you can draw different sized rectangles by writing, for example:

```
rectangle (30 , 80)
```