Kojo/Scala

Andrea Sterbini – sterbini@di.uniroma1.it

Kojo learning environment: Scala language + turtles + math/geometry

Kojo IDE:

- HTML "stories" to guide you

- Code editor

- 2D canvas (turtle-based)

- 2D math and geometry cartesian canvas (Geogebra):
    - lines, points, segments, angles, areas, formulas, plots, …

- Arithmetic exercises
- Music player (with Midi instruments)

Other: Arduino programming (with an arduino driver/interface)

Scala programming language

Scala <u>functional</u> programming language:

- compiles to the Java VM   (and can import Java libs)
- <u>simpler readable syntax</u>   (e.g.   object method parameter)
- functions + OOP
  - anonymous functions, map/filter/…
  - simplified definition of classes (no need for getters/setters)
- preferred immutable structures vs. mutable:
  - immutable (val) vs. mutable (var) variables
  - imm. collections are copied while mutable can be changed
  - method arguments are always immutable
- operator overloading

Programming style

Single-threaded?	YES	(Concurrency? YES)
Procedural?	YES	(Lazy evaluation? YES)

<u>Functional?</u> <u>YES!</u>

Statically typed:
- complex abstract types   - metaprogramming
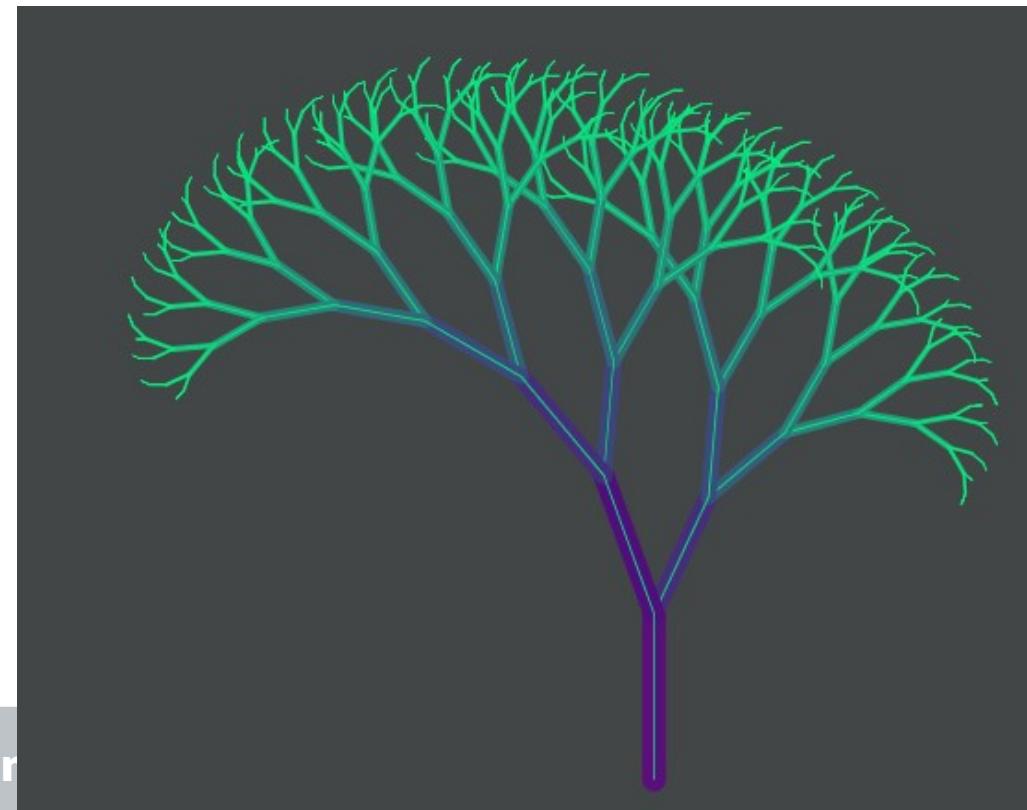
Data types:
- Objects + Classes (+ Singletons + Structs)
- sequential/parallel collections
- list-based operations   (map/filter/...)
- anonymous functions   (code blocks)
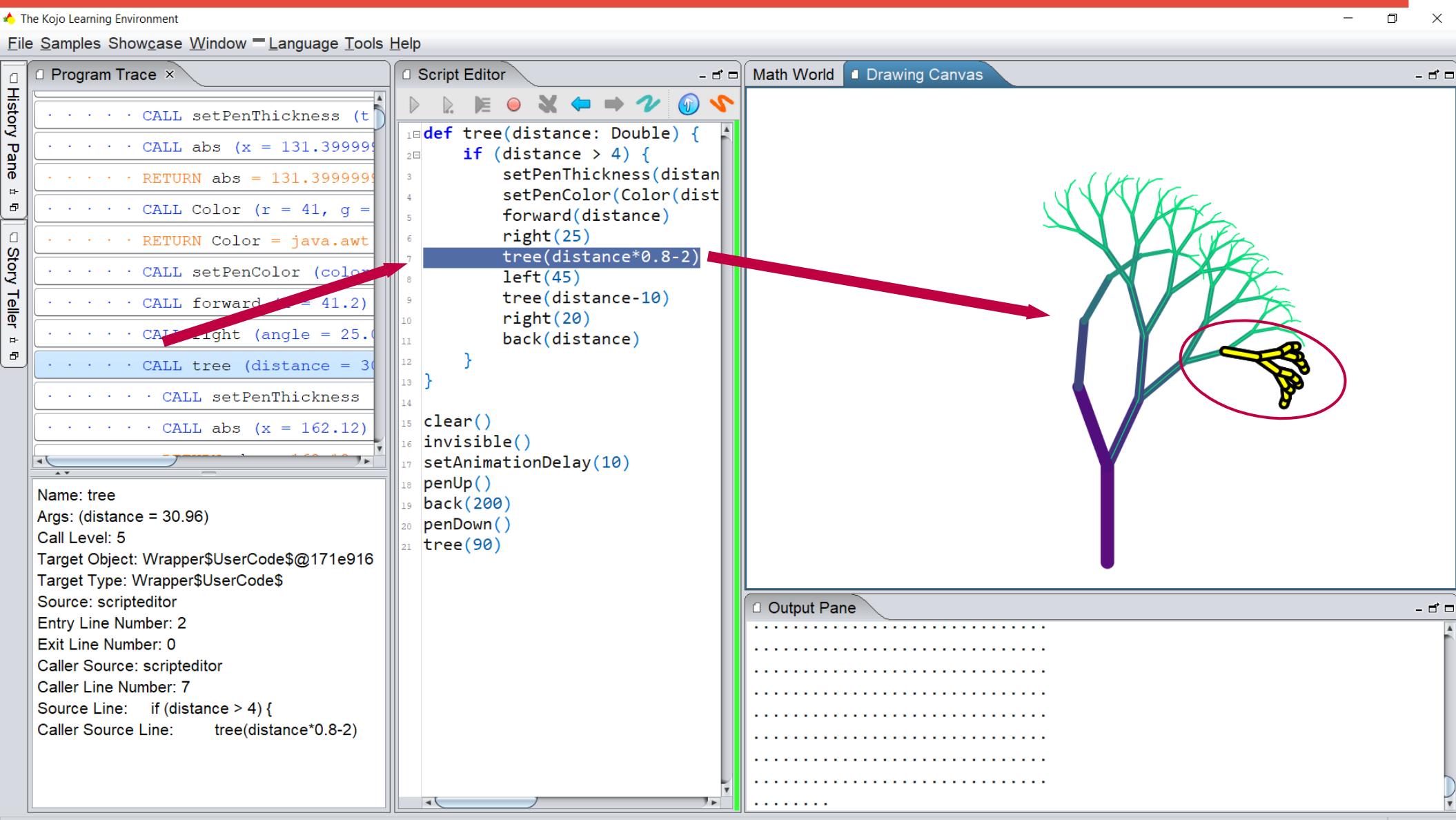
Demo 1: recursive tree by turtle graphics (as usual)

```
clear()
invisible()
setAnimationDelay(10)
penUp()
back(200)
penDown()
tree(90)
```

```
def tree(distance: Double) {
  if (distance > 4) {
    setPenThickness(distance/7)
    setPenColor(
        Color(distance.toInt, math.abs(255-distance*3).toInt, 125))
    forward(distance)
    right(25)
    tree(distance*0.8-2)
    left(45)
    tree(distance-10)
    right(20)
    back(distance)
  }
}
```

Good debugger with recursion trace/visualization

Kojo/Scala

Tree2 = recursive transformations of a rectangle
rotate(-25°) + scale(72%) + brightness(+10%)

tree(10)

tree(2)

tree(1)

Demo 2: recursive tree = recursive pictures + graphic transformations

```
// we start with a square
def Square = Picture {
  repeat (4) {
    forward(100)
    right()        // default 90°
  }
}

// a stem is a distorted black square
def stem = scale(0.13, 1) *
  penColor(noColor) *
  fillColor(black) -> Square
```

```
def drawing(n: Int): Picture = {
  if (n < 2)
    stem
  else
    GPics(stem,
      trans(0, size + 10) * brit(0.1) -> Gpics(
          rot(-25) * scale(0.72)  -> drawing(n-1),
          rot(  50) * scale(0.55)  -> drawing(n-1)
        )
    )
}
clear()
setBackground(Color(255, 170, 29))
invisible()
val pic = trans(0, -300) -> drawing(10)
draw(pic)
```

Demo 3
choose the correct article for an italian word

Type: definite/indefinite (determinativo/indeterminativo)
Gender: male/female
Number: singular/plural

1) deduce the word gender from final char
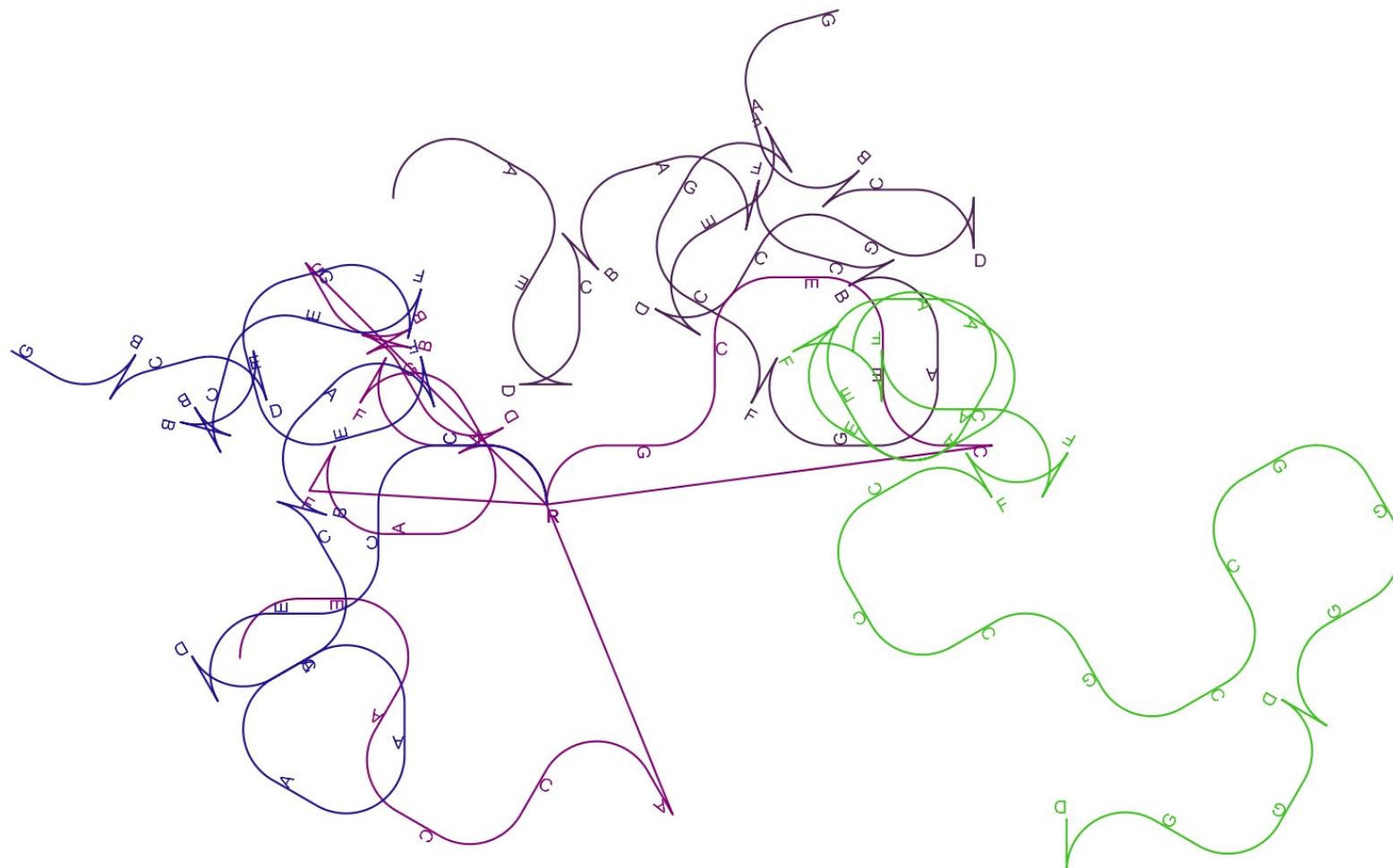
2) select proper gender/number from final char

3) handle normality and exceptions (here for ind. male sing. only)
   - starts with vowel   → "un"
   - starts with consonant   → "un"
   - starts with 2 special vowels ('ia', 'ie', 'io', 'iu')   → "uno"
   - starts with 1 or 2 special consonants   → "uno"
      ("x", "y", "z", "gn", "pt", "ps", "pn", "sc", "sf", "sq", "st")

# Demo 4: music transposition and art
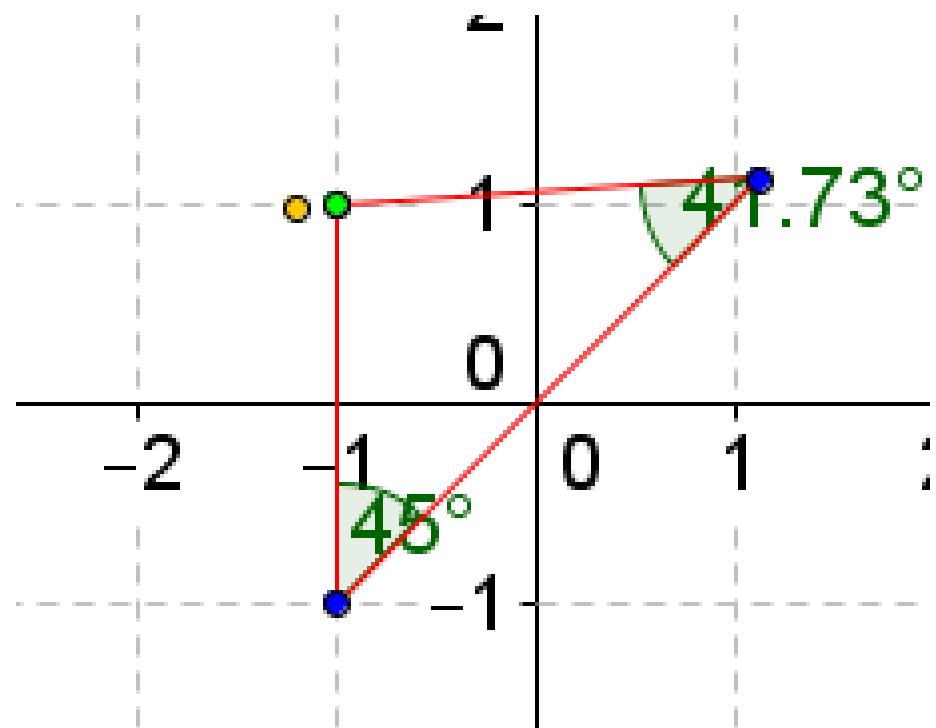
- Many voices/instruments   - With arrangement
- Transposed to a different key/mode (major/minor)
- Each voice produces a coloured path (each note a segment)

Just use turtle graphics with the Mw class to create lines and angles

```
val t = Mw.turtle(-1, 0)
t.showExternalAngles()
t.forward(-2)
t.right(45)
t.forward(3)
t.moveTo(-1,0)
```


Full GeoGebra with spreadsheet

Not (yet?) possible to get all properties FROM Geogebra elements

Arithmetic exercises

Sum, Multiplication, Division, Subtraction

Demo

DEMO