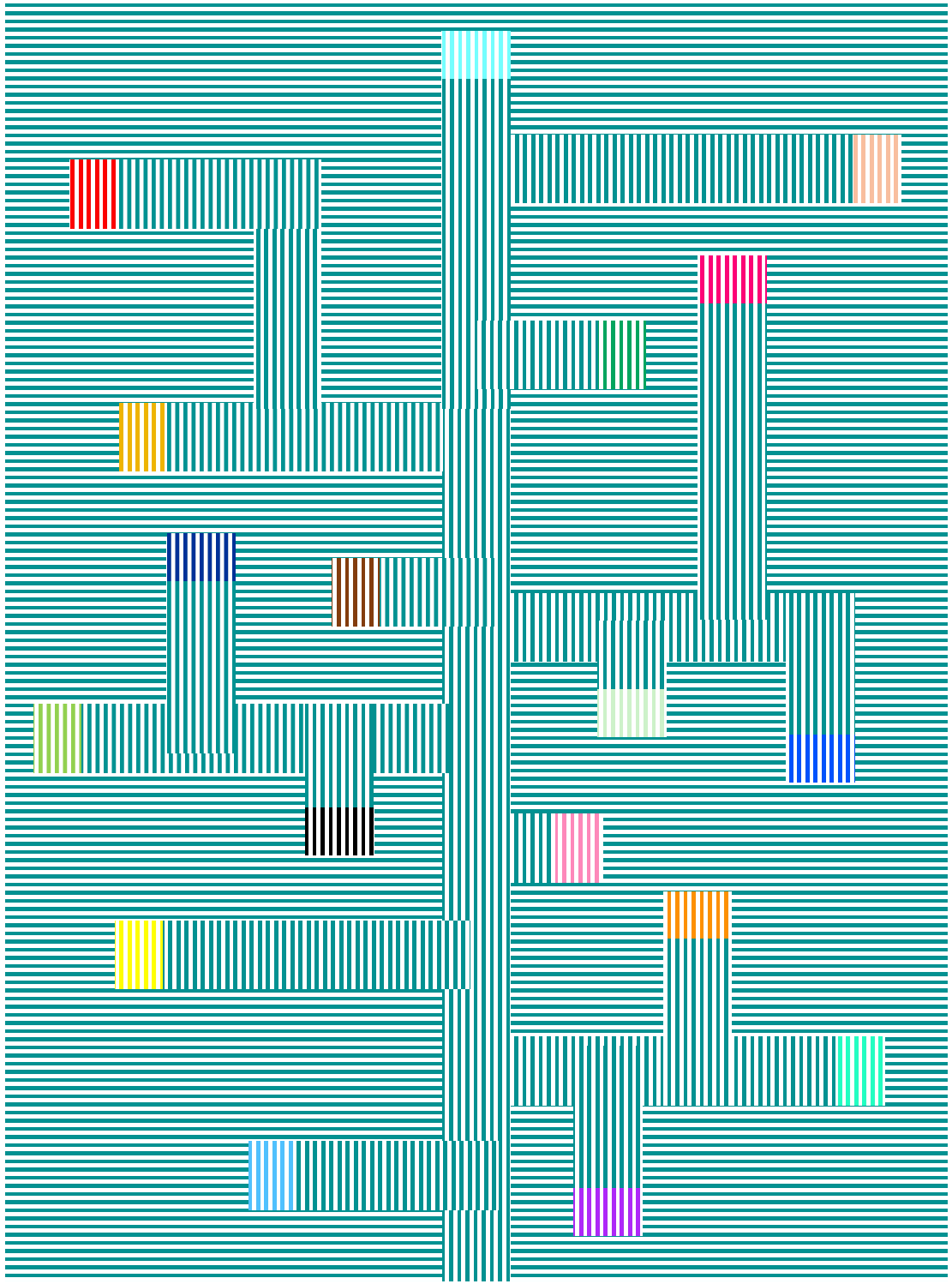
Digital Humanities Lecture 12 May 12 2025

Mario Verdicchio
Cecilia Scatturin

A bit of self-promotion
Un po’ di autopromozione

Milan Design Week 2023
Milano, FuoriSalone 2023

Milan Design Week 2024
Milano, FuoriSalone 2024



The original design
on PowerPoint
Il disegno originale
su PowerPoint

The truth / La verità

- The design was done “by hand” inside a digital environment.
- The curator wanted a computer code to make the work look “cooler”.
- A code was written afterwards and presented as the origin of the design.

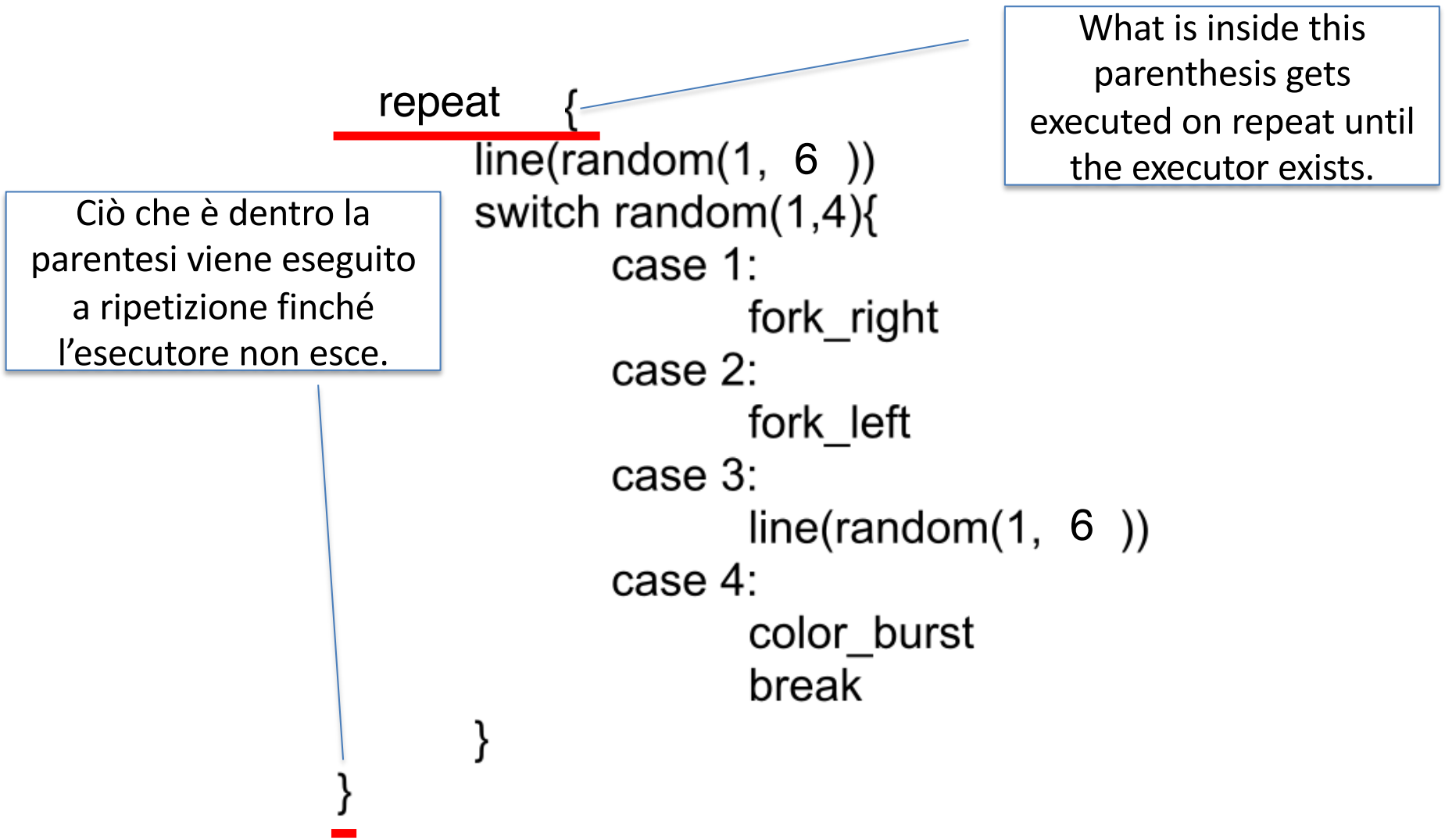
- Il progetto è stato realizzato “a mano” all’interno di un ambiente digitale.
- Il curatore desiderava un codice informatico per rendere l’opera più “cool”.
- Un codice è stato scritto successivamente e presentato come l’origine del progetto.

The Tree of Decisions / L'albero delle decisioni
Mario Verdicchio

```
while (true){
        line(random(1,100))
        switch random(1,4){
                case 1:
                        fork_right
                case 2:
                        fork_left
                case 3:
                        line(random(1,100))
                case 4:
                        color_burst
                        break
        }
}
```

For today's class /
Per la lezione di oggi

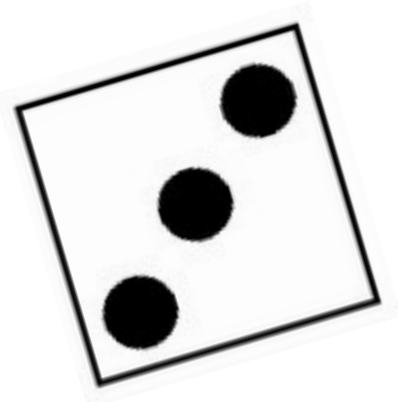
```
repeat {
        line(random(1, 6 ))
        switch random(1,4){
                case 1:
                        fork_right
                case 2:
                        fork_left
                case 3:
                        line(random(1, 6 ))
                case 4:
                        color_burst
                        break
        }
}
```

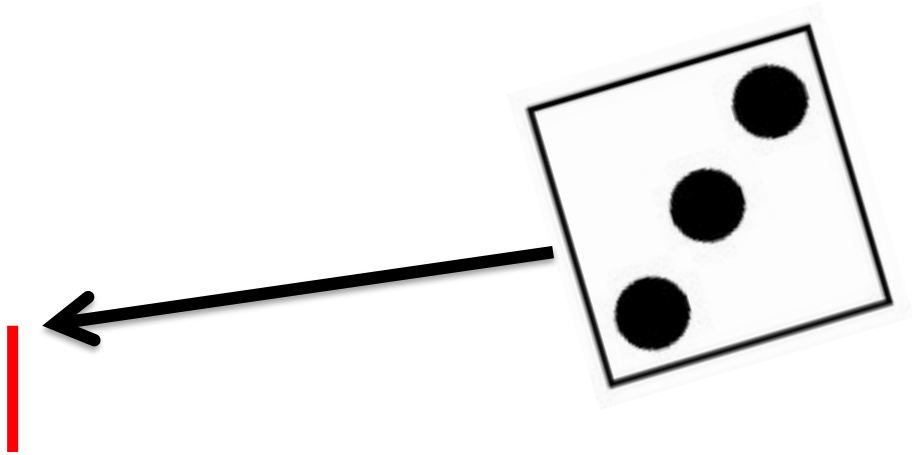

repeat {
line(random(1, 6))
switch random(1,4){
case 1:
fork_right
case 2:
fork_left
case 3:
line(random(1, 6))
case 4:
color_burst
break
}
}
What is inside this parenthesis gets executed on repeat until the executor exists.
Ciò che è dentro la parentesi viene eseguito a ripetizione finché l'esecutore non esce.

```
repeat    {
        line(random(1,  6  ))
        switch random(1,4){
                case 1:
                        fork_right
                case 2:
                        fork_left
                case 3:
                        line(random(1,  6  ))
                case 4:
                        color_burst
                        break
        }
}
```

First thing to do:
draw a line of random length (min 1 max 6)

Prima cosa:
disegnare una linea di lunghezza casuale (min 1 max 6)

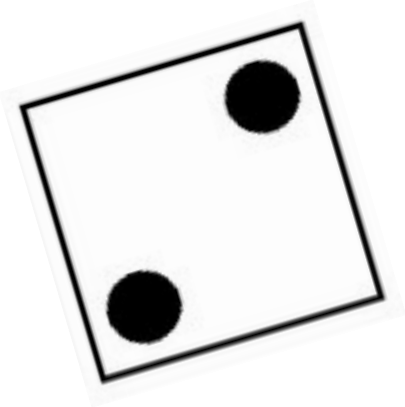




```
repeat    {
        line(random(1,  6  ))
        switch random(1,4){
                case 1:
                        fork_right
                case 2:
                        fork_left
                case 3:
                        line(random(1,  6  ))
                case 4:
                        color_burst
                        break
        }
}
```

Then, there is a random choice between 1 and 4, which determines what to do next.

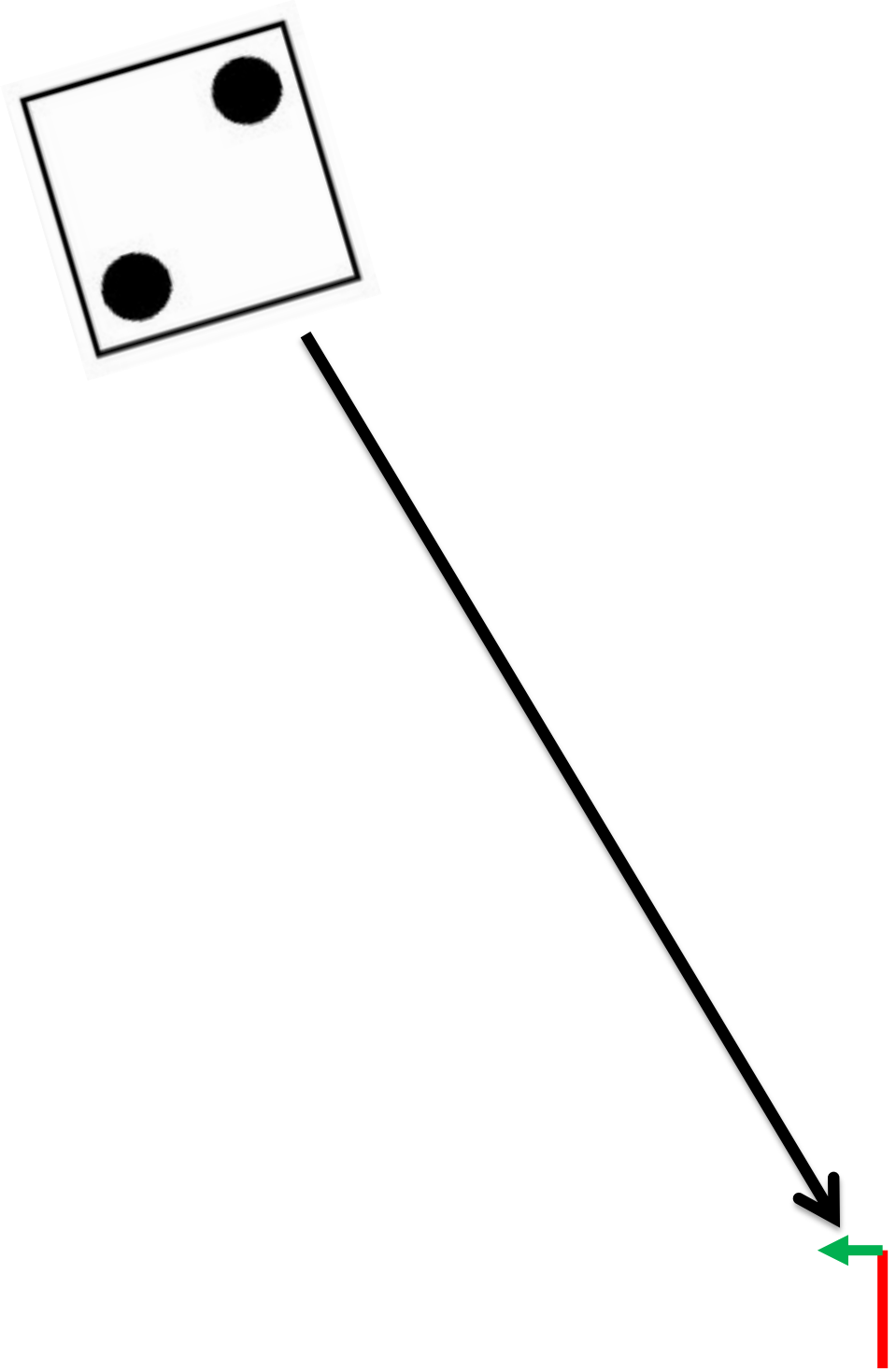
Poi c'è una scelta a caso da 1 a 4, che determina la prossima azione.



```
repeat    {
        line(random(1,  6  ))
        switch random(1,4){
                case 1:
                        fork_right
                case 2:
                        fork_left
                case 3:
                        line(random(1,  6  ))
                case 4:
                        color_burst
                        break
        }
}
```

Mettiamo che esca 2. Allora viene creata una copia del processo (programma in esecuzione) che procede verso sinistra.

Suppose 2 is the case. Then a copy of the process (program in progress) is created and it proceeds towards the left.



The new process starts / Il nuovo processo inizia

```
repeat    {
        line(random(1,  6  ))
switch random(1,4){
        case 1:
                fork_right
        case 2:
                fork_left
        case 3:
                line(random(1,  6  ))
        case 4:
                color_burst
                break
        }
}
```

What is inside this parenthesis gets executed on repeat until the executor exists.

First thing to do: draw a line of random length (min 1 max 6)

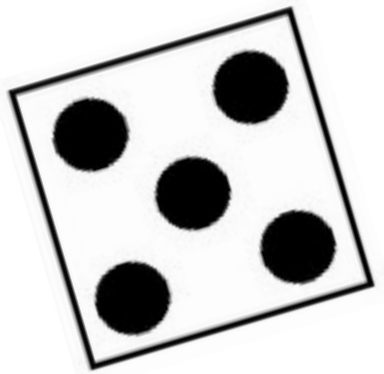
Prima cosa: disegnare una linea di lunghezza casuale (min 1 max 6)

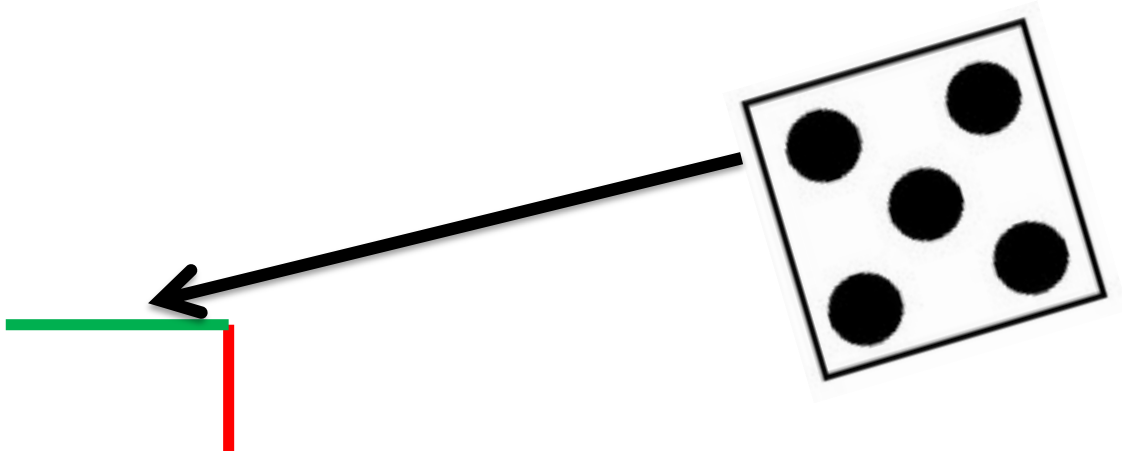
Ciò che è dentro la parentesi viene eseguito a ripetizione finché l'esecutore non esce.

```
repeat    {
        line(random(1,  6  ))
        switch random(1,4){
                case 1:
                        fork_right
                case 2:
                        fork_left
                case 3:
                        line(random(1,  6  ))
                case 4:
                        color_burst
                        break
        }
}
```

First thing to do:
draw a line of random length (min 1 max 6)

Prima cosa:
disegnare una linea di lunghezza casuale (min 1 max 6)

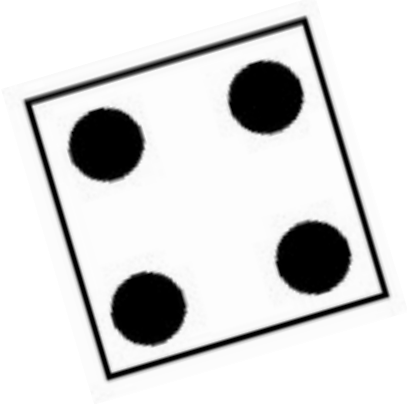




```
repeat    {
        line(random(1,  6  ))
        switch random(1,4){
                case 1:
                        fork_right
                case 2:
                        fork_left
                case 3:
                        line(random(1,  6  ))
                case 4:
                        color_burst
                        break
        }
}
```

Then, there is a random choice between 1 and 4, which determines what to do next.

Poi c'è una scelta a caso da 1 a 4, che determina la prossima azione.

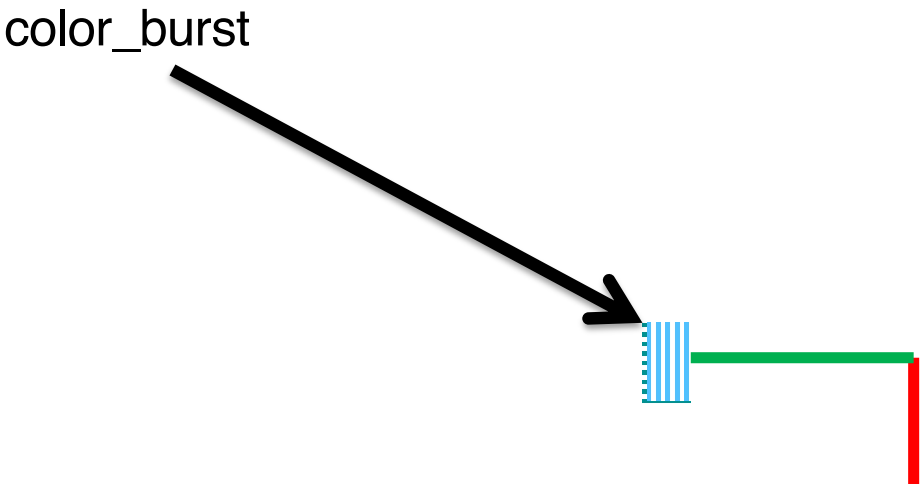


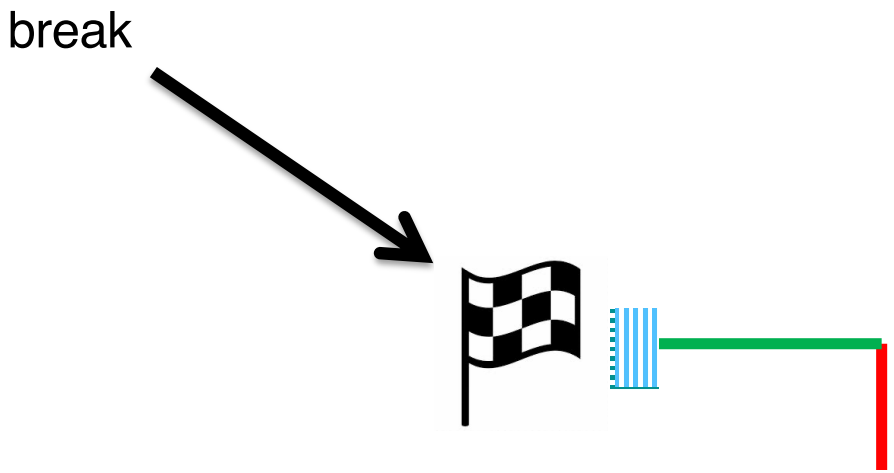
```
repeat    {
        line(random(1,  6  ))
        switch random(1,4){
                case 1:
                        fork_right
                case 2:
                        fork_left
                case 3:
                        line(random(1,  6  ))
                case 4:
                        color_burst
                        break

                }
        }
```

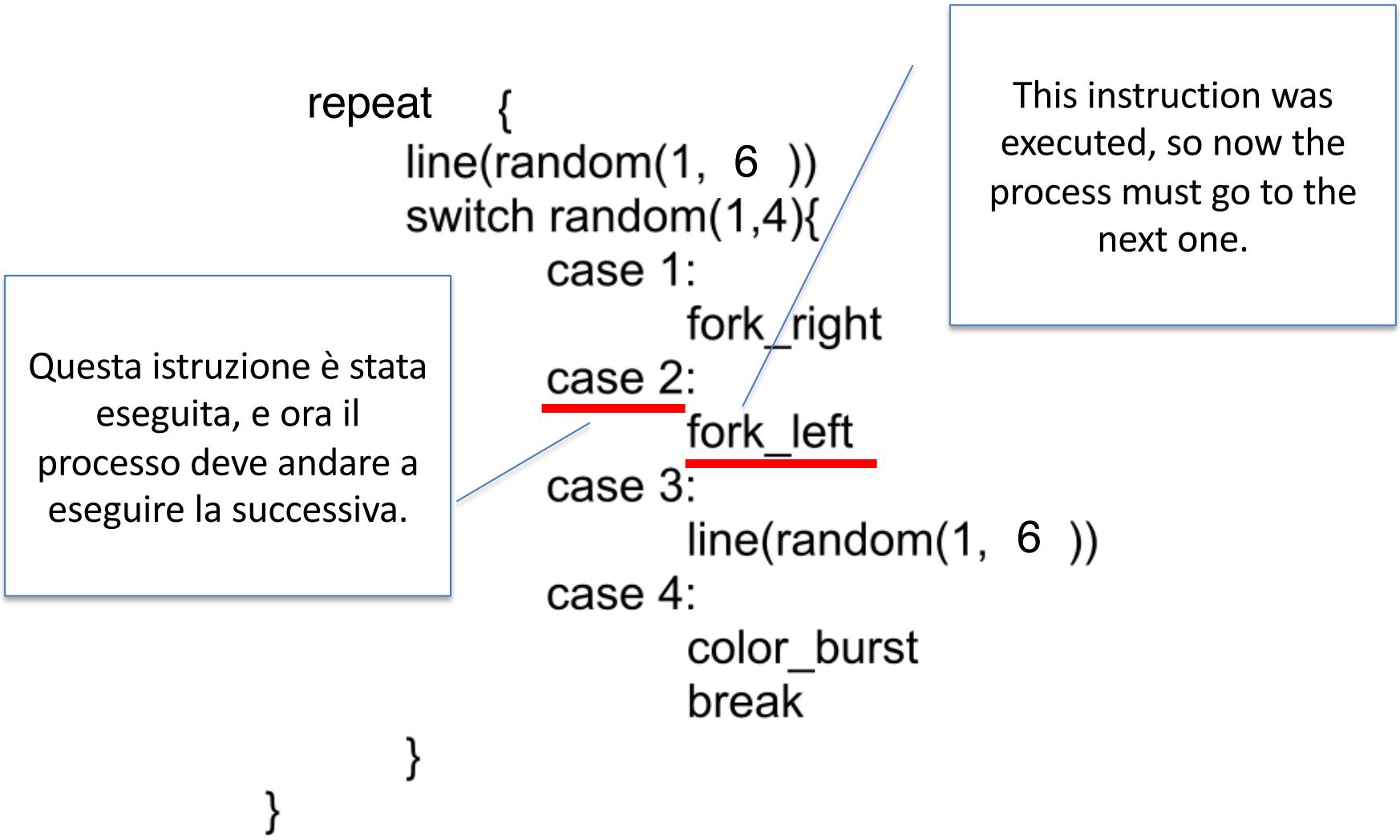
Mettiamo che esca 4. Allora viene eseguita l'istruzione color_burst e poi break, che fa uscire dal blocco "repeat".

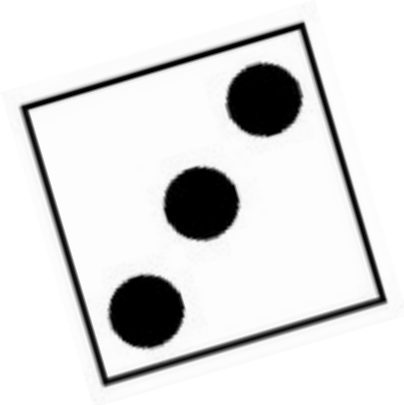
Suppose 4 is the case. Then color_burst is executed and then break, which makes the process exit the "repeat" block.


color_burst


break

Let’s get back to the first process.
/
Torniamo al primo processo.


repeat {
line(random(1, 6))
switch random(1,4){
case 1:
fork_right
case 2:
fork_left
case 3:
line(random(1, 6))
case 4:
color_burst
break
}
}
This instruction was executed, so now the process must go to the next one.
Questa istruzione è stata eseguita, e ora il processo deve andare a eseguire la successiva.

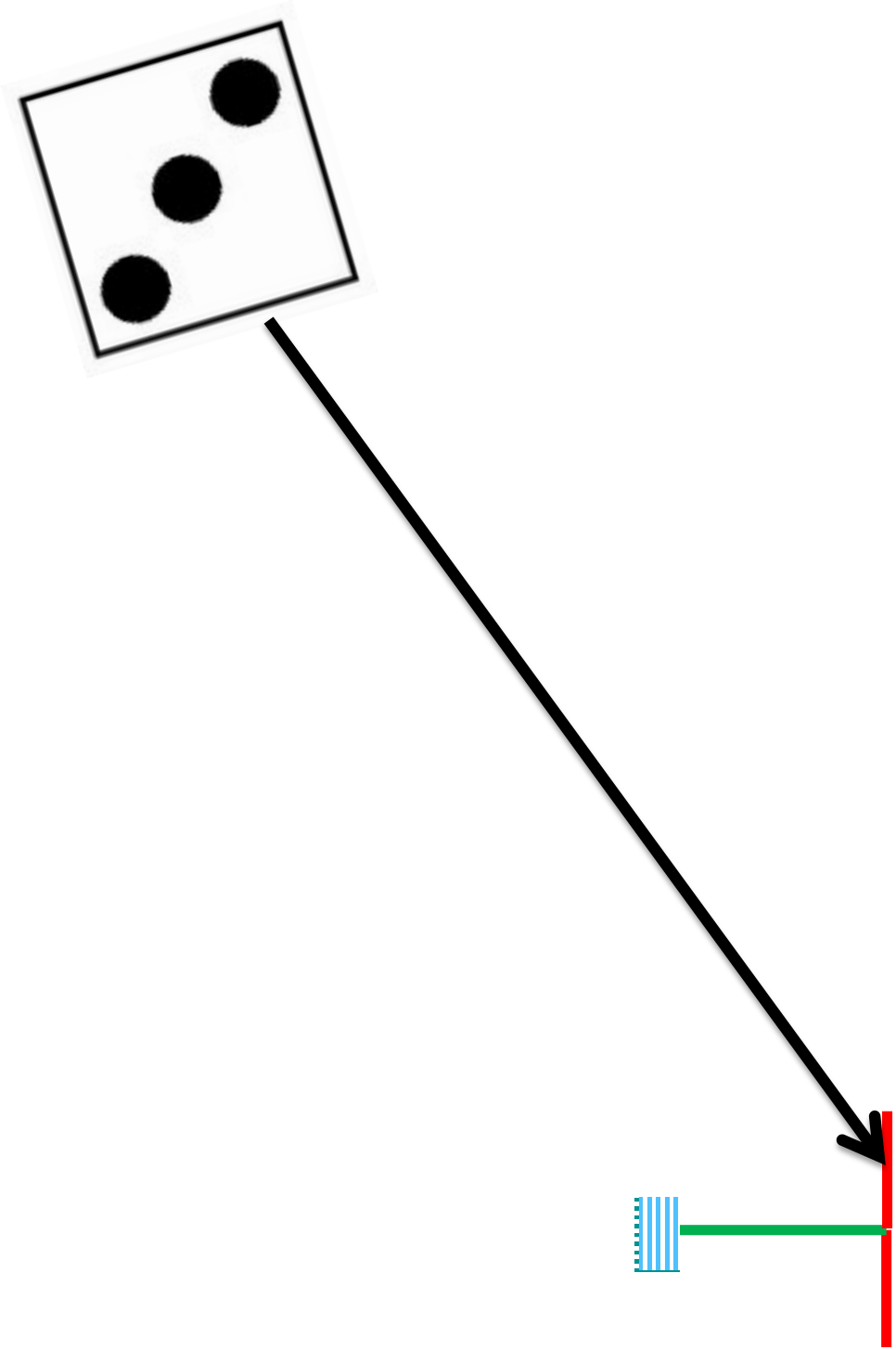


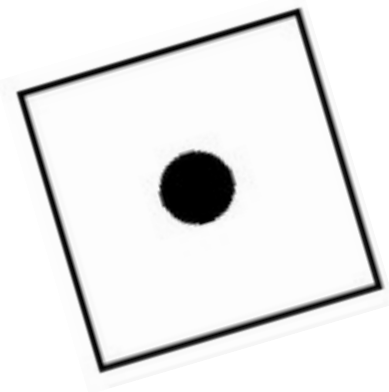
```
repeat    {
        line(random(1,  6  ))
        switch random(1,4){
                case 1:
                        fork_right
                case 2:
                        fork_left
                case 3:
                        line(random(1,  6  ))
                case 4:
                        color_burst
                        break

                }
        }
```

Finché non viene eseguita la break il processo rimane nel blocco “repeat”, quindi torna all’inizio ed esegue l’istruzione “line”. Supponiamo che venga 3.

Until the break is executed, the process remains in the “repeat” block, so it goes back to the beginning and executes the “line” instruction. Suppose 3 is the case.

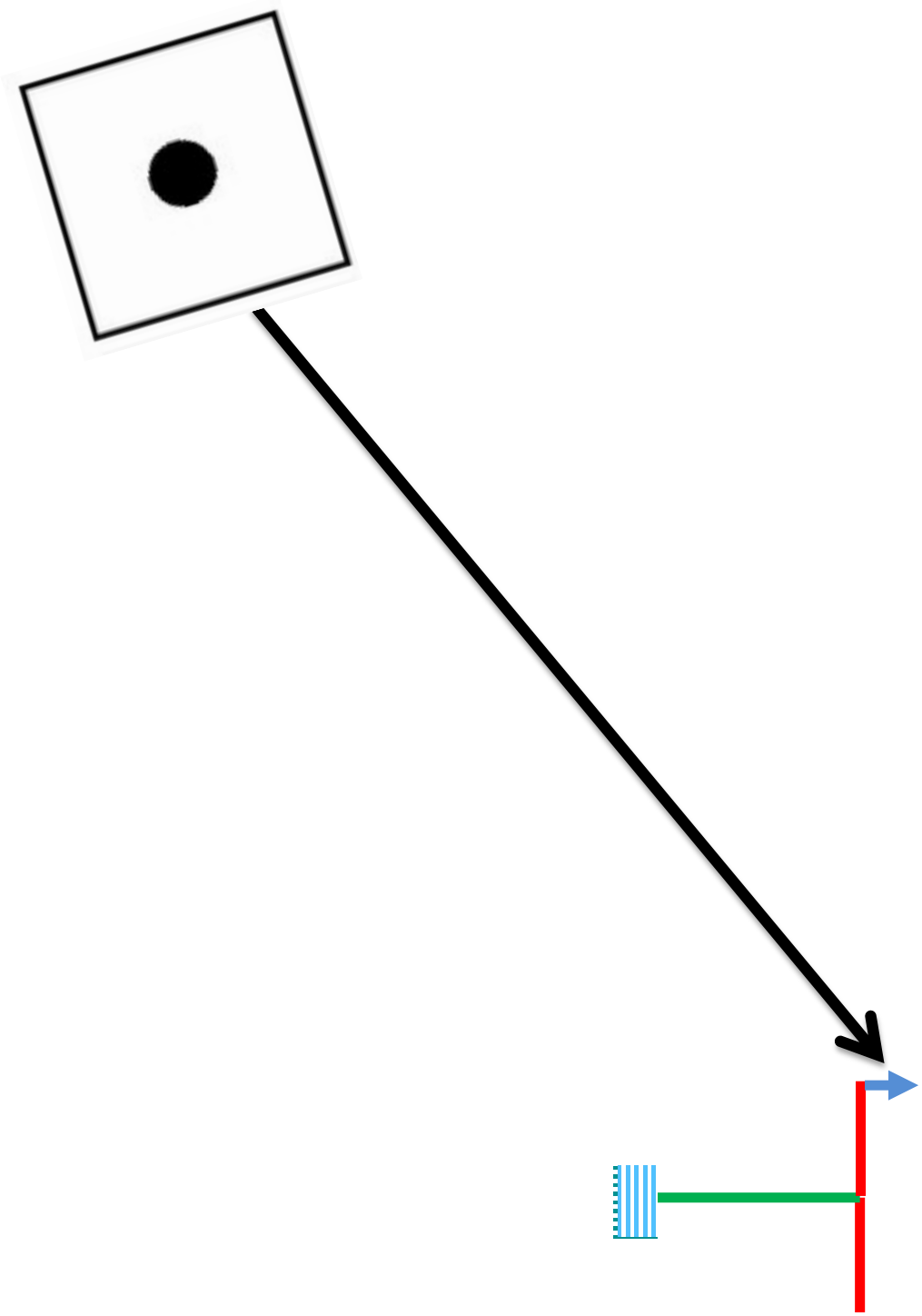




```
repeat    {
        line(random(1,  6  ))
        switch random(1,4){
                case 1:
                        fork_right
                case 2:
                        fork_left
                case 3:
                        line(random(1,  6  ))
                case 4:
                        color_burst
                        break
        }
}
```

Supponiamo questa volta la scelta a caso tra 1 e 4 cada su 1. "fork_right" crea un nuovo processo che va a destra.

Let's suppose this time the random choice between 1 and 4 gives 1. "fork_right" creates a new process that goes right.

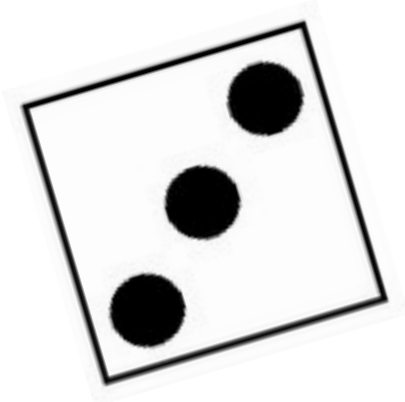


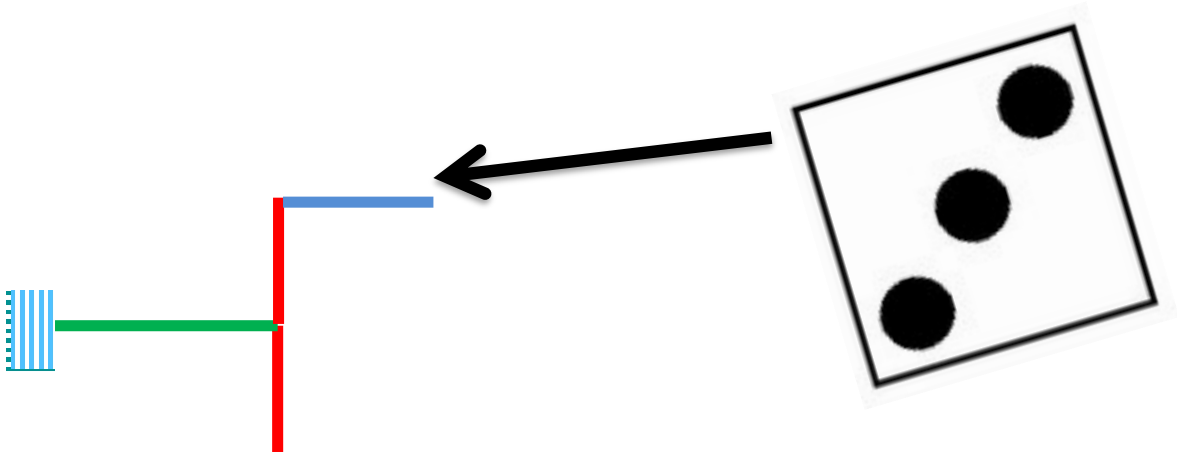
The new process starts / Il nuovo processo inizia

```
repeat    {
        line(random(1,  6  ))
        switch random(1,4){
                case 1:
                        fork_right
                case 2:
                        fork_left
                case 3:
                        line(random(1,  6  ))
                case 4:
                        color_burst
                        break
        }
}
```

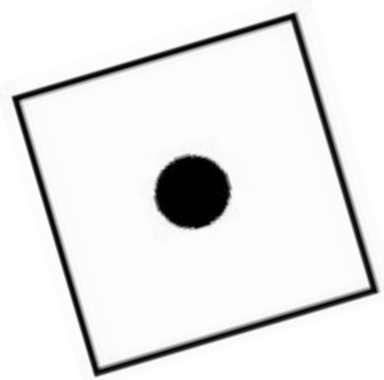
First thing to do:
draw a line of random length (min 1 max 6)

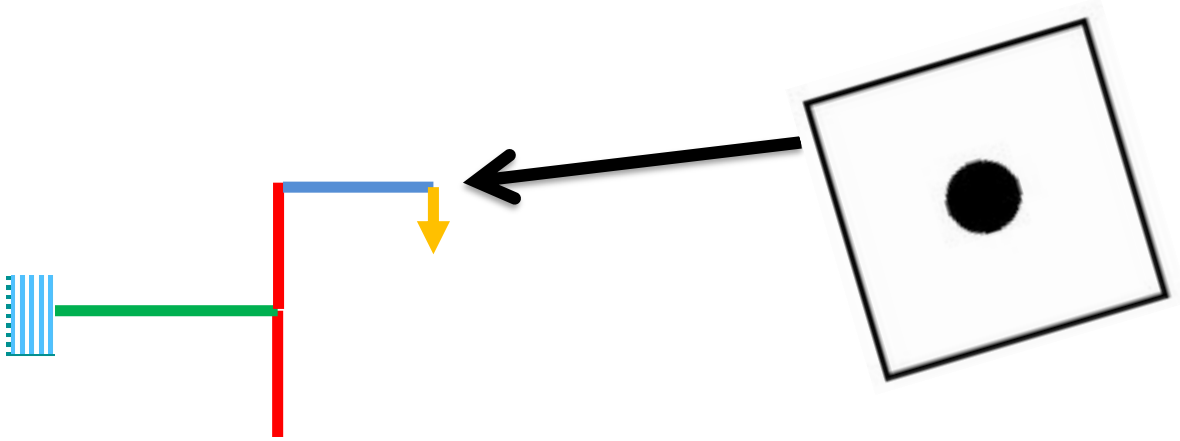
Prima cosa:
disegnare una linea di lunghezza casuale (min 1 max 6)





```
repeat    {
        line(random(1,  6  ))
        switch random(1,4){
                case 1:
                        fork  right
                case 2:
                        fork_left
                case 3:
                        line(random(1,  6  ))
                case 4:
                        color_burst
                        break
        }
}
```

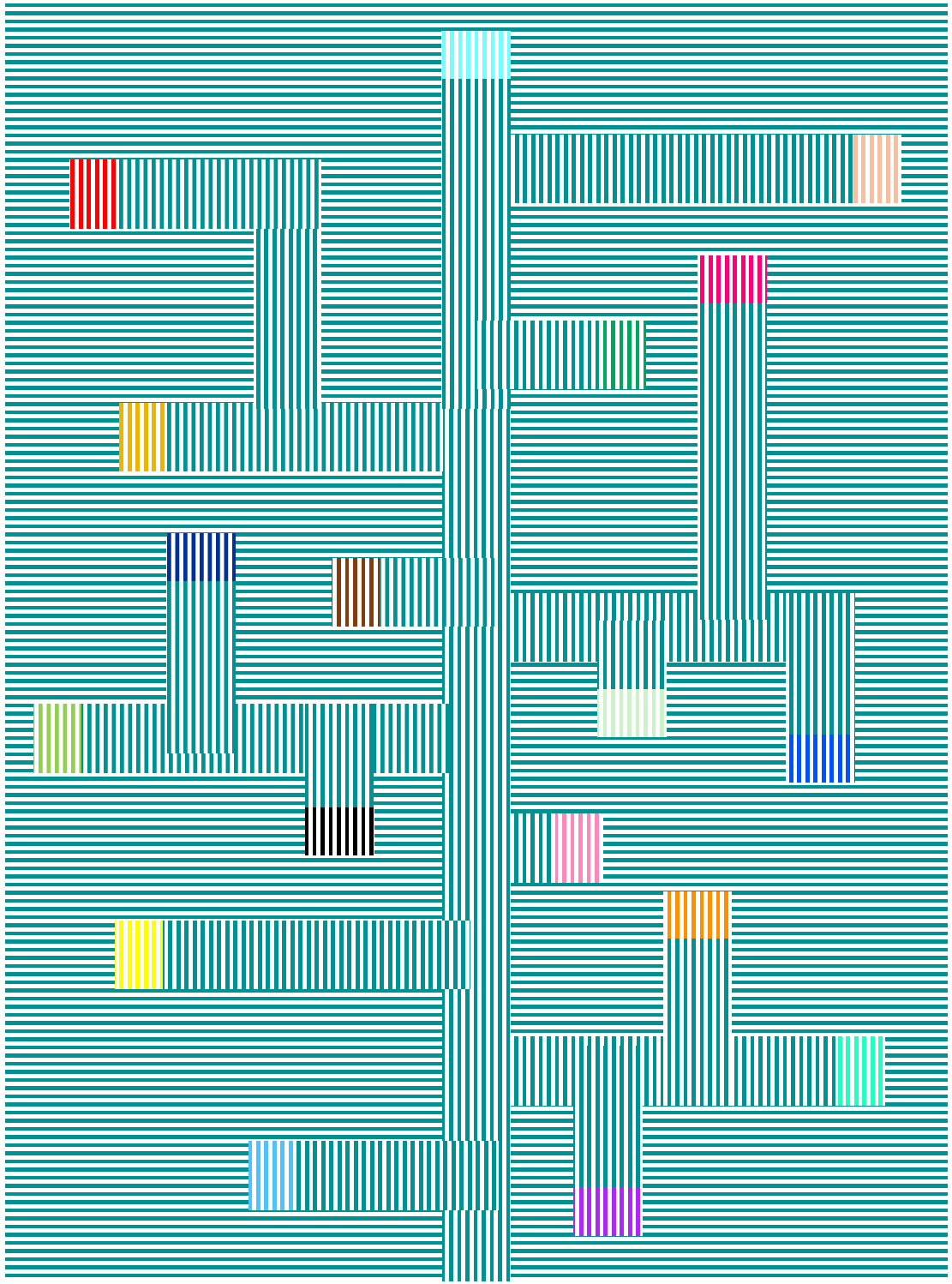




And so on, until all processes terminate.

/

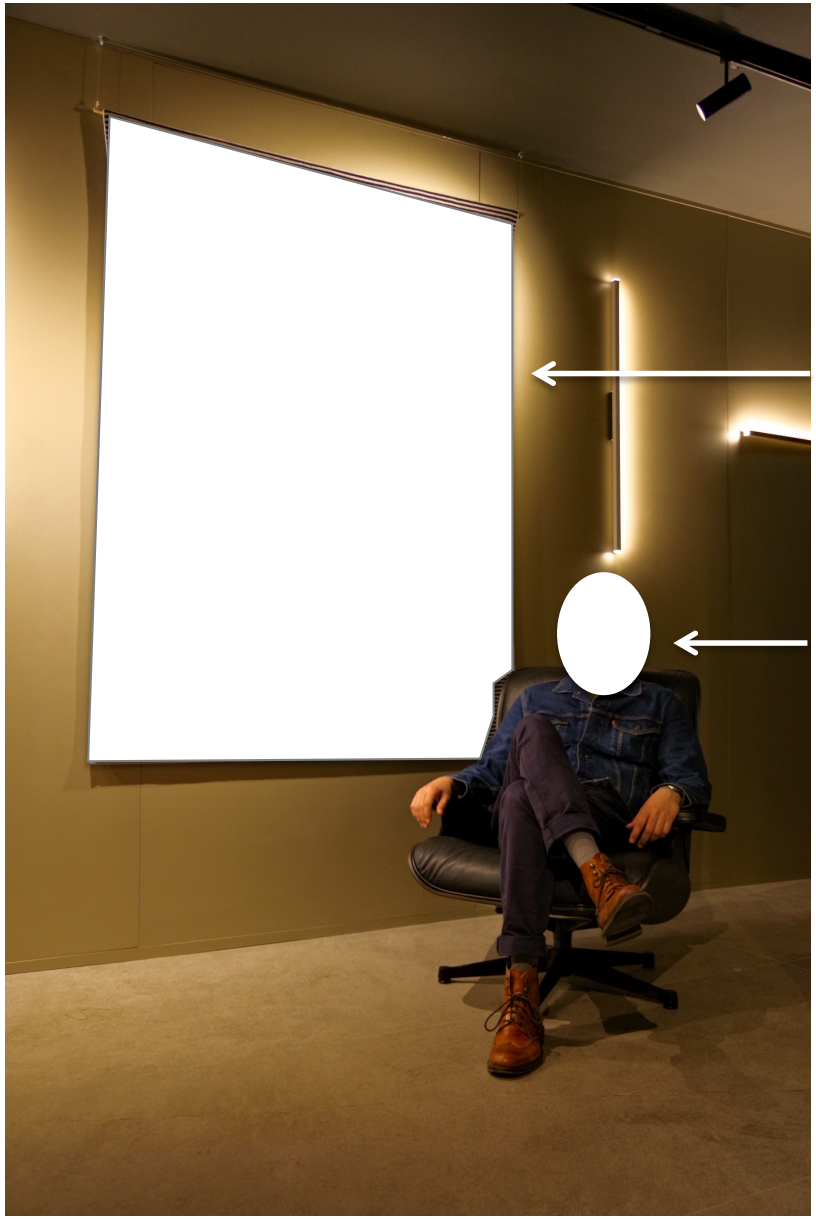
E così via, finché tutti i processi non terminano.



Not all details are specified in the program. You will discover things as you try and execute it.

/

Non tutti i dettagli sono specificati nel programma. Li scoprirete mentre provate ad eseguirlo.



Insert your design here.
Il tuo design qui.

Insert your face here.
La tua faccia qui.