

1 AD2: River Simulation

1.1 Objetivo

O objetivo da AD2 é complementar as tarefas que ficaram faltando na AD1:

- Implemente uma interface gráfica, para alterar os parâmetros da simulação, e que permita executar todas as tarefas necessárias, fornecendo:
 - a configuração inicial do rio,
 - número de ciclos,
 - a semente para o gerador de números aleatórios,
 - nome do arquivo, etc...

A saída do programa também deve ser efetuada usando componentes de interface apropriados.

- Teste o seu código completamente. Utilize o unittest:

<https://docs.python.org/3/library/unittest.html>

A melhor forma de fazer isso é usando configurações pequenas, por exemplo, com três animais e testando todas as possibilidades. Outra forma, é empregando sempre a mesma semente, porém o resultado pode ser diferente, dependendo da versão do python, por ex. 2 e 3. Deve ser entregue um arquivo RiverTest.py que importa a classe River e efetua os testes.

Procure testar todos os aspectos da sua implementação...

A interface implementada deve ser a mais adequada possível aos requisitos da simulação. Não há um formato fixo. Use a sua criatividade e bom senso...

2 Interface

A interface é bastante simples: bastam 3 entries, um button, dois radio-buttons e alguns labels.

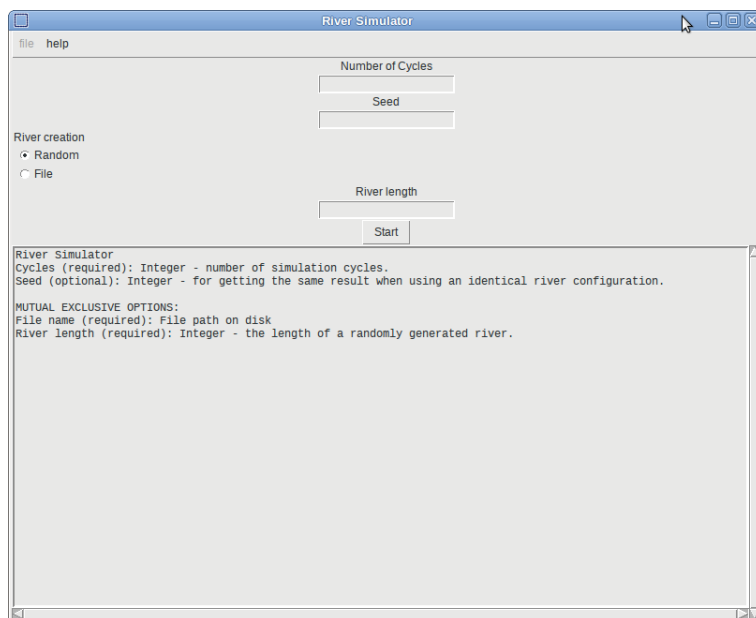


Figura 1: Interface do projeto.

O resultado da simulação é exibido em uma área de Text, com dois scrollbars: horizontal e vertical.

Há também um cascade menu para exibir mensagens de ajuda (help) e salvar a saída da simulação em um arquivo.

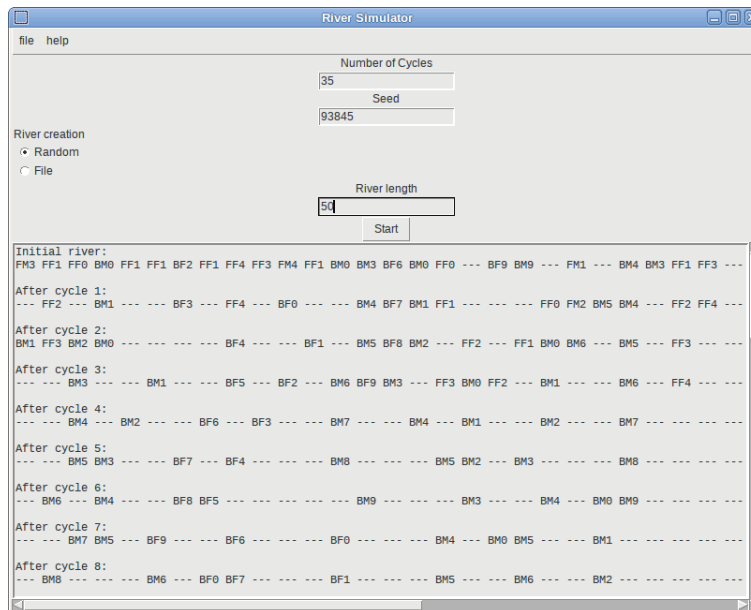


Figura 2: Simulação.

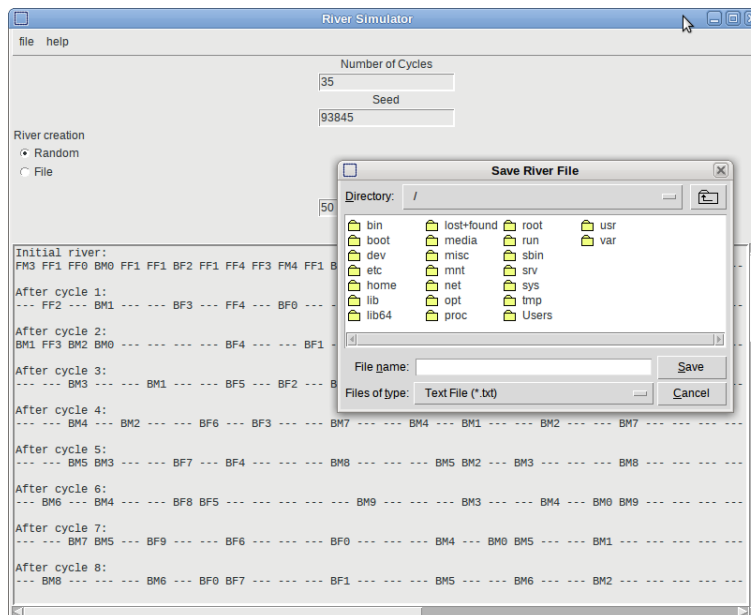


Figura 3: Arquivo de saída.

3 Testes

Os testes estão num arquivo RiverTest.py, por exemplo, como o listado abaixo:

```
#!/usr/bin/env python
# coding: UTF-8
#
## @package RiverTest
#
# Unit test for certain aspects of the behavior of
# RiverSimulator.
#
# @author Paulo Cavalcanti
# @date 13/02/2018
# @see https://docs.python.org/3/library/unittest.html
#

from River import *
import sys
import unittest

class RiverTest(unittest.TestCase):
    ##
    # Seed for controlling random number generator.
    #
    SEED = 19580427

    def test_FF3_FF4_BF4(self):
        river = River("RiverTest.txt")

        # "FF3 FF4 BF4" -> FF4 dies old

        # FF4 — BF5

        msg1 = "BF5 — ——" # BF5 moved to the right (FF4 moved
                           # to the left (eaten) or did not move)
        msg2 = "— — BF5" # FF4 moved to the left and BF5 did not move
```

```

msg3 = "---- BF5 ----" # BF5 moved to the left and FF4 moved to right
msg4 = "FF4 —— BF5" # BF5 and FF4 did not move
msg5 = "FF4 BF5 ----" # BF5 moved to the left and FF4 did not move
msg6 = "BF5 FF4 ----" # FF4 moved to the right and BF5 to the right
msg7 = "---- FF4 BF5" # FF4 moved to the right and BF5 did not move

self.assertEqual("FF3 FF4 BF4", str(river))

print (river)
river.updateRiver()
print (river)

self.assertIn("BF5", str(river))
self.assertNotEqual(str(river),"---- —— ----")

# seven possibilities
self.assertTrue(msg1 == str(river) or
                msg2 == str(river) or
                msg3 == str(river) or
                msg4 == str(river) or
                msg5 == str(river) or
                msg6 == str(river) or
                msg7 == str(river))

river.updateRiver()
print (river)
self.assertIn("BF6", str(river))
self.assertNotIn("FF", str(river))

# FF4 dies old
# BF6 can be in any of the three cells

msg1 = "BF6 —— ----"
msg2 = "---- —— BF6"
msg3 = "---- BF6 ----"

# three possibilities
self.assertTrue(msg1 == str(river) or

```

```

        msg2 == str(river) or
        msg3 == str(river))

    # create a river with length 5
    # we want to repeat the same choices.
    river = River(5, RiverTest.SEED)
    print(river)
    river.updateRiver(True)
    print(river)
    # python 2 and 3 give different results.
    self.assertTrue(str(river) == "BM7 BM0 BM0 — BF3" or str(river) \
                    == "— — BM9 — —")

def main():
    test = RiverTest()
    test.test_FF3_FF4_BF4()

if __name__=="__main__":
    unittest.main()

```

```

[cascavel:~/html/python/labs2/RiverSimulation] p3 RiverTest.py
FF3 FF4 BF4
— — BF5
— — BF6
End of Simulation.
FM4 BF9 — BM8 —
— — BM9 — —
End of Simulation.
.

```

```

Ran 1 test in 0.041s

```

```

OK

```