Presentation of our created ICT resource

SimulP200 is an ICT resource whose acronym is derived from its full name: ''Polonium 200 radioactive decay Simulation chessboard'' [23]. This ICT resource was developed by our team using the Python language. This language is placed online under a free license (https://www.python.org/psf/). Indeed, after having installed this software on five pupils' laptops, in addition to the teacher's laptop, its interface offers the possibility to create code files, in this case, the code of SimulP200. Once properly written by our team and saved in laptop computers, each file become ready to use because it has the same distinctive logo and a name it has been given.

Software Source Code of SimulP200

```
from tkinter import Tk, Canvas, Frame, BOTH, Label, StringVar
from random import randint
from time import sleep
import numpy as np
dt=30
class Example(Frame):
    def __init__(self, n, m, size):
        super().__init__()
        self.n = n
        self.m = m
        self.size = size
        self.R = []
        self.canvas = Canvas(self, width= 10 + self.n*self.size + 200 , height=10 + self.m*self.size + 100)
        self.t=0
        self.T=np.arange(0,5520)
        self.N=(1000*np.exp(-0.001*self.T)).astype('int')
        self.initUI()
    def initUI(self):
        self.master.title("")
        self.status_label=StringVar()
        self.status_label.set("Simulation de la décroissance radioactive du polonium 200 (demi-vie:690s)\nChaque carreau orange est un noyau radioactif. En se désintégrant, il vire au vert")
        w3 = Label(self.master, textvariable=self.status_label,font=("Helvetica", 24))
        w3.pack()
```

```
        self.time_label=StringVar()
        self.time_label.set("t = %ds"%self.t)
        self.reds_label=StringVar()
        self.reds_label.set("N(t) =% d"%self.N[self.t])
        w = Label(self.master, textvariable=self.time_label,font=("Helvetica", 24))
        w.pack()
        w2 = Label(self.master, textvariable=self.reds_label,font=("Helvetica", 24))
        w2.pack()
        self.pack(fill=BOTH, expand=1)
        x = 10
        for i in range(self.n):
            y = 10
            for j in range (self.m):
                self.R.append(self.canvas.create_rectangle(x, y, x + self.size , y + self.size,
outline="#f00", fill="#fb0"))
                y = y + self.size
            x = x + self.size
        self.canvas.pack(fill=BOTH, expand=1)
    def change (self):
        self.t=self.t+1
        self.time_label.set("t = %ds"%self.t)
        self.reds_label.set("N(t) = %d"%self.N[self.t])
        if(len(self.R)>self.N[self.t]):
            p = randint(0,len(self.R)-1)
            self.canvas.itemconfigure(self.R[p], fill="#0f0")
            self.R.pop(p)
root = Tk()
ex = Example(40, 25, 20)
for i in range(5520):
    ex.after(dt+i*dt, ex.change)
root.mainloop()
```