**Exercise 6 - back to the desert island**

*General instructions:*

This task is done and returned in the same way as the previous exercise 5 - the same instructions and rules apply - as well as the same practice for the final result, i.e. the video clip and . py file recovery.

**Description of the task**

After their last vacation ended in a shipwreck, friends Ernesti and Kernesti decide to try the vacation again. On the previous vacation trip, Ernest and Kernest's vacation trip was not successful due to a shipwreck, but on the way they luckily escaped to a deserted island, from where they finally got back to the "people's world", so to speak. Most obviously, future skilled software experts played a big role in the rescue work .

Ernesti and Kernesti fondly remember the deserted island and especially its friendly monkeys. How nice it was there - until you got out after being saved!

That's how the friends decide to go on another vacation trip, but this time the destination is exactly the same deserted island that they ended up on almost by accident.

The trip to the island is successful without any problems, and once there, Ernesti and Kernesti start an "active vacation", meaning they always sunbathe in the mornings and spend the afternoons wandering in the forest where the island's monkeys are buzzing. They also - albeit at a small risk - occasionally go swimming in the shark-infested sea. During these refreshing swimming trips, they discover that all the sharks in the sea are enormous. They don't see any sharks less than one meter in diameter. But, as brave former coders, Ernesti and Kernesti don't care about the danger of sharks any more. After all, both of them have already brought millions and millions of euros to society in tax revenue alone, and besides, they are already retired!

***A plan for a safe swimming pool***

But, one day a huge hairy shark tried to eat both Ernest and Kernest in the same swimming trip and even with the same lunge . They only narrowly escaped the situation thanks to their quick reflexes. The matter left both of them thinking a little, and so they decide to do something about it.

So they are planning to make a swimming pool in the middle of the island, where the water would come from a narrow ditch about a meter wide from the sea. And, it would be safe to swim in this pool, because big sharks (more than two meters in diameter) can't get into the pool from the narrow ditch and disturb the vacationers floating there. Old engineers can find solutions to every issue! Thus,

-create a deserted island with a sandy island - preferably sand yellow - in the middle of a blue ocean. Illustrate the inside of the island as you wish by outlining a swimming pool, size 20x60 meters and additionally Draw two ditches into it (one is "Ernest's ditch" and the other " Kernest's ditch"), both one hundred meters long and one meter wide (i.e. the whole 100x1 meter), extending into the pool -from the pool to the sea.

The shared dream is that in the future Ernesti could sunbathe on the left edge of the shark-free swimming pool, watching the water flow through his own ditch into the swimming pool and back. And accordingly, a similar structure would be on Kernest's side.

-define the pool as a matrix with a size of 20x60. This pool is a natural depression on the island, but of course it is empty because no water from the sea can get into it. Therefore, set the number values of the matrix to zeros to reflect this. If water enters the pool in the future, the numerical values of the matrix will be greater than zero.

-define the ditches accordingly as matrices with sizes of 100x1. Of course, these ditches do not exist on the island yet, so they are only plans for ditches at this stage of the project. Thus, "fill" these ditches with sand, i.e., mark the matrix values as "ones" at each point to illustrate that there is still "sand" in the ditch at a certain point.

SEA 
SAND 

-when these steps are done smoothly, run the command i\_suppose\_i\_have\_earned\_so\_much\_ points ( 1 )

***Procurement of labor needed for construction***

When Ernesti and Kernesti have planned the swimming pool and outlined the ditches through which water can be obtained, they wonder how on earth they will get the necessary long ditches dug from the sea to the pool? Both of them remember that third-year engineering student who lives next door to their own home, who would dig a little ditch like that in a hurry! But when he's not here right now...what advice?

Kuva, joka sisältää kohteen luonnos, Piirrokset, piirros, clipart

Kuvaus luotu automaattisesti

Finally, Kernest has a flash; but those monkeys of ours! Let's ask them to dig that ditch! And that's how the matter can be handled...

-create a functionality for Ernest that fetches an idle monkey from the forest, gives it a shovel and guides this monkey to Ernest's ditch (between points A\_e and B\_e ) to some point. Use the threading.Thread () command to create this functionality . However, don't let the monkey go digging just yet at this stage...

Kernesti and Ernesti recall their successful programming careers in working life. The secret of their personal success and their continuous salary increase cycle was that they always worked intensively and quickly for three minutes of high-quality software, and after that they always remembered to take a quarter-hour break to refresh themselves and do something fun, and only after that they always continued working. The same principles certainly also apply to digging a ditch, at least partially, which is why...

-creates a functionality for digging, where when t.start () or a similar type of command is given, the monkey digs from the point where he is "ditching", "pixel by pixel", i.e. "one element of the ditch matrix at a time" towards the sea. Practically assign the matter to the "ditch matrix" in such a way that whenever a ditch is dug at a certain point, the element corresponding to the ditch matrix at that point becomes zero, and for the sake of illustration, the appropriate sound effect is heard at the same time. Remember to illustrate the location of the monkey along the ditch if you haven't already done so

The important thing is to set the digging speed to one "meter" per second for the first second. Because of the monkey's fatigue, it takes two seconds to dig the next "meter". The physiology of the arm muscles of monkeys is much the same as that of a software engineer's brain - i.e. at first they always manage quickly, but natural fatigue phenomena will inevitably come into play over time. So, when digging a ditch, digging the next meter takes twice as long, i.e. four seconds, and so on...

-try sending AND assigning one monkey to dig Ernest's ditch. If the monkey goes to the starting point correctly AND then starts digging correctly, run the command i\_suppose\_i\_have\_earned\_so\_much\_ points ( 2 )

-next, make a similar functionality for Kernest for fetching monkeys lounging in the forest and guiding them to the side of the ditch to work.

***More together***

Both software development and ditch digging go best as teamwork. Because of that…

-adds a function that allows Ernesti to add a new monkey to dig a ditch at any time. When putting a new monkey to work, it may be worth considering that the monkey should be placed in a place where the ditch has not yet been dug. If you accidentally make a monkey dig a ditch in a place where a ditch has already been dug, it will only be correspondingly deeper, i.e. the ditch will change from, for example, "zero" deep to "minus one". Or, if in poorly organized ditch digging "minus the depth of one" more ditch is dug, of course it becomes "minus the depth of two"

-add a similar function to Kernest .

Kuva, joka sisältää kohteen luonnos, piirros, Piirrokset, viivapiirros

Kuvaus luotu automaattisesti

-observe on the island how the digging of the ditch progresses, i.e. how the "ditch matrix" visually changes on the screen as the monkeys dig their ditches towards the sea.

-when you can smoothly employ monkeys to dig the ditch in such a way that as the ditch progresses both Ernest's and Kernest's ditch is clearly formed, run the command

i\_suppose\_i\_have\_earned\_so\_much\_ points ( 3 )

***Optimal use of resources***

When digging a ditch seems to be successful in terms of basic operations, Ernesti and Kernesti think about how on earth digging a ditch should be done as smartly as possible? How many monkeys should be put to work, where?

In order to investigate this, Ernesti and Kernesti are kicking up the ditches that have been dug mainly for practice, freeing the monkeys from their work, i.e....

-to fill the "ditch" make a function that changes the values of the elements of the ditch matrices back to "ones"

-create a function for Ernest that first randomly places one monkey between the pool and the sea, and immediately puts him to work. After that, create some logic that cleverly places the next monkey on the side of the ditch after one second, then the next one after one second, and so on. Repeat this up to ten monkeys!

-Create a corresponding function for Kernest

-when these are done, run the command

i\_suppose\_i\_have\_earned\_so\_much\_ points ( 4 )

***The goal is a full swimming pool***

Of course, in the end, Ernesti and Kernesti want to compete to see whose ditch will be finished first, if the work starts on both ditches at the same time. So…

-fill the "ditches" as usual

-creates a phenomenon for the sea using the threading.Thread () principle, which constantly monitors whether the seawater has access to the ditches. If there is, seawater always fills the ditch as the water has a path to enter. Whenever more water enters the ditch, produce the corresponding sound effect.

-also creates a phenomenon for the swimming pool with the threading.Thread () principle, which fills it if the water reaches it from the ditch. When the pool is full, create the appropriate sound effect as follows:

-when the swimming pool is full, indicate with a low beep lasting 10 seconds if Ernest's ditch was the one that was finished first or if Kernest's ditch was the first to be finished, indicate this accordingly with a high-pitched beep lasting 10 seconds.

-when all these are done, run the command

i\_suppose\_i\_have\_earned\_so\_much\_ points ( 5 )

Appendix A

# template

import tkinter as tk

import winsound

import time

window = tk . Tk ()

window . title ( " Exercise 5" )

window . geometry ( "700x700" )

# add five buttons to the top line of the window

decoration = tk . Label ( window , text = "" ). grid ( row = 0 , column = 0 )

point\_button =[ ]

for i in range ( 5 ):

    button\_temp = tk . Button ( window , text = " Points : " + str ( i + 1 ), padx = 40 )

    button\_ temp . grid ( row = 0 , column = i + 1 )

    point\_button . \_ append ( button\_temp )

def i\_suppose\_i\_have\_earned\_so\_much\_points ( amount\_of\_points ):

    for i in range ( 5 ):

        point\_button [ i ]. configure ( bg = 'gray' )

    time . sleeping ( 1 )

    for i in range ( amount\_of\_points ):

        point\_button [ i ]. configure ( bg = ' green ' )

        winsound . Beep ( 440 + i \* 100 , 500 )

# example ...

i\_suppose\_i\_have\_earned\_so\_much\_ points ( 3 )

window . mainloop ()