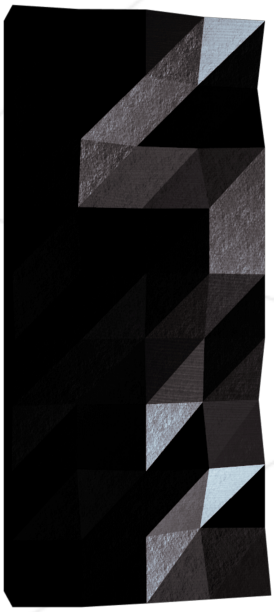


OCaml Pool - Rush 01

Instant Tama

42 pedago pedago@42.fr
kashim vbazenne@student.42.fr
nate alafouas@student.42.fr

Abstract: This is the subject for rush01 of the OCaml pool.
The main theme of this day is to use a graphic library with OCaml in order to create a basic Tamagotchi like game.

Contents

Chapter I

Ocaml piscine, general rules

- Every output goes to the standard output, and will be ended by a newline, unless specified otherwise.
- The imposed filenames must be followed to the letter, as well as class names, function names and method names, etc.
- Unless otherwise explicitly stated, the keywords `open`, `for` and `while` are forbidden. Their use will be flagged as cheating, no questions asked.
- Turn-in directories are `ex00/`, `ex01/`, ..., `exn/`.
- You must read the examples thoroughly. They can contain requirements that are not obvious in the exercise's description.
- Since you are allowed to use the `OCaml` syntaxes you learned about since the beginning of the piscine, you are not allowed to use any additionnal syntaxes, modules and libraries unless explicitly stated otherwise.
- The exercices must be done in order. The graduation will stop at the first failed exercice. Yes, the old school way.
- Read each exercise FULLY before starting it ! Really, do it.
- The compiler to use is `ocamlopt`. When you are required to turn in a function, you must also include anything necessary to compile a full executable. That executable should display some tests that prove that you've done the exercice right.
- Remember that the special token `";;"` is only used to end an expression in the interpreter. Thus, it must never appear in any file you turn in. Anyway, the interpreter is a powerfull ally, learn to use it at its best as soon as possible !
- The subject can be modified up to 4h before the final turn-in time.
- In case you're wondering, no coding style is enforced during the `OCaml` piscine. You can use any style you like, no restrictions. But remember that a code your peer-

evaluator can't read is a code she or he can't grade. As usual, big fonctions is a weak style.

- You will NOT be graded by a program, unless explictly stated in the subject. Therefore, you are afforded a certain amount of freedom in how you choose to do the exercises. Anyway, some piscine day might explicitly cancel this rule, and you will have to respect directions and outputs perfectly.

- Only the requested files must be turned in and thus present on the repository during the peer-evaluation.

- Even if the subject of an exercise is short, it's worth spending some time on it to be absolutely sure you understand what's expected of you, and that you did it in the best possible way.

- By Odin, by Thor ! Use your brain !!!

Chapter II

Day-specific rules

- You MUST not push your graphical library in the repository but provide a clever Makefile rule to install it instead.
- You are in a functional programming pool, so your coding style MUST be functionnal (Except for the side effects for the input/output). I insist, your code MUST be functional, otherwise you'll have a tedious defence session.
- For each exercise of the day, you must provide sufficient material for testing during the defence session. **Every functionnality that can't be tested won't be graded!**
- You must provide an author file at the root of your git with your login in it, as usual...

Chapter III

Foreword

Here lies a pretty song based on the famous Instant Karma from John Lennon
Instant Tama's gonna get you
Gonna knock you right on the head
You better get yourself together
Pretty soon you're gonna be dead
What in the world you thinking of
Laughing in the face of love
What on earth you tryin' to do
It's up to you, yeah you

Instant Tama's gonna get you
Gonna look you right in the face
Better get yourself together darlin'
Join the OCaml race
How in the world you gonna see
Laughin' at fools like me
Who in the hell d'you think you are
A super star
Well, right you are

Well we all cried out
Like the staff and the other students
Well we all cried out
Ev'ryone come on

Instant Tama's gonna get you
Gonna knock you off your feet
Better recognize your brothers
Ev'ryone you meet
Why in the world are we here
Surely not to live in pain and fear
Why on earth are you there
When you're ev'rywhere
Come and get your share

Well we all cried out
Like the staff and the other students
Well we all cried out

Well we all cried out
Like the staff and the other students
Well we all cried out

Well we all cried out
Like the staff and the other students
Well we all cried out
Like the staff and the other students
Well we all cried out
Like the staff and the other students
Well we all cried out
Like the staff and the other students

Chapter IV

Mandatory Part

In this part you must provide sufficient functionnality to demonstrate your almighty power in OCaml.

- `Your program must draw a wibbily wobbly timey wimey creature on the main screen`
- `You must also draw basic meters : health, hygiene, energy, happyness (you're free to draw it as you like but it must remain consistent with the concept.)`
- `You must also draw basic button/area to allow the end user to select an action to perform : EAT, THUNDER, BATH, KILL`

Your pet must obey the following rules:

- `Health is initially setted to 100 and is depleted by 20 each time an action other than EAT is performed and depleted by 1 each second`
- `Energy is initially setted to 100 and is depleted by 10 each time an actionother than THUNDER is performed`
- `Hygiene is initially setted to 100 and is depleted by 20 each time an EAT action is performed`
- `Happyness is initially setted to 100 and is depleted by 5 each time an action is performed`

Here is the list of the actions and their side effects:

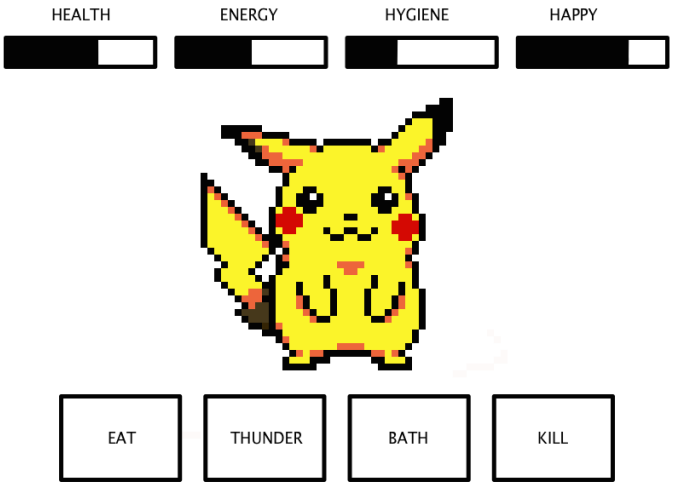
- `EAT : Health + 25, Energy - 10, Hygiene - 20, Happiness + 5`
- `THUNDER : Health - 20, Energy + 25, Happiness - 20`
- `BATH : Health - 20, Energy - 10, Hygiene + 25, Happiness + 5`
- `KILL : Health - 20, Energy - 10, Happiness + 20`

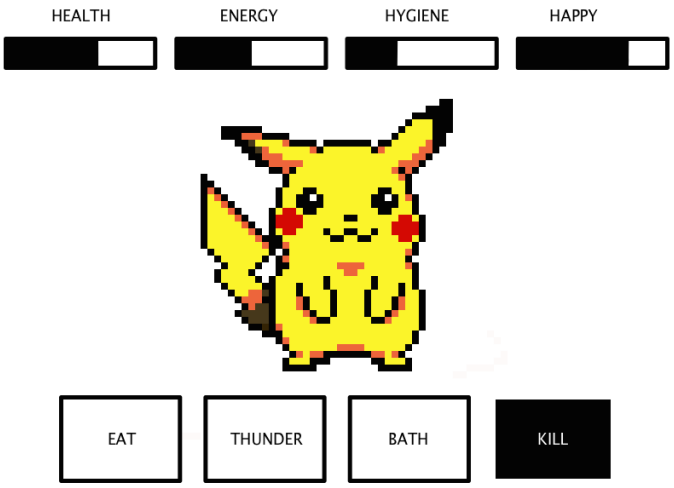
Also if any of the stats of your pet reach the value 0, your pet is instantly killed dramatically with a GAME OVER.

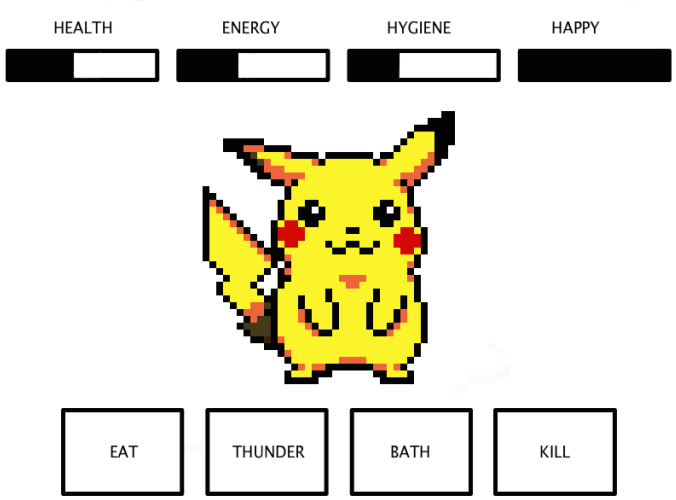
You should also provide an efficient way to save the state of your game into a file named **save.itama** and load it correctly at the next launch. (It would be wise to use the notions from the d09 of your OCaml pool...)

You're free to use any graphic library you like to achieve this rush.

Below are some screenshots of a relatively beautiful majestic as fuck Instant Tama.






HEALTH
ENERGY
HYGIENE
HAPPY
EAT
THUNDER
BATH
KILL

Chapter V

Bonus Part

You're free to provide any functionnality you want as bonus. Here are some of the most common functionnnality :

- `New actions : SLEEP, SING, DANCE... but you must not alter the mandatory actions!`
- `Super Gangsta Graphics with vertex shaders, tiles, and all the mega features of you library`
- `Sonor effects when an action is performed`

WARNING : if you program crash at any time or throw an unhandled exception, your work won't be graded, even if the crash comes from a BONUS.