PokemOZ

Jérémie Melchior

1. Project description

Hello there! Welcome to the world of POKEMOZ! My name is JAY! People call me the POKEMOZ PROF! This world is inhabited by creatures called POKEMOZ! For some people, POKEMOZ are pets. Others use them for fights. Myself… I study POKEMOZ as a profession.
Right! So you are a student who has learnt how to represent different situations on computers and now we are going to use that knowledge to create a game. As most of you already know the real game, we are going to briefly describe it and we will make some adjustments to the rules and to the game play. You will have to simulate POKEMOZ trainers, POKEMOZ and fights against wild POKEMOZ and other POKEMOZ trainers. The game starts at your home and your goal is to reach mine.
Students! Your very own POKEMOZ legend is about to unfold! A world of dreams and adventures with POKEMOZ awaits! Let's go!

But first, let's define the game components:

- **POKEMOZ** are animals either wild or owned by a trainer.
- POKEMOZ **trainers** are human characters that use POKEMOZ to fight against each other and against wild POKEMOZ.
- The **map** is a mix of roads and tall grass areas.
- **Roads** are used by trainers to go from points to points. There are no wild POKEMOZ walking on roads. However other trainers can used roads.
- **Tall grass** is an area where wild POKEMOZ live.

JAY: Hey! Wait! Don't go out! It's unsafe! Wild POKEMOZ live in tall grass! You need your own POKEMOZ for your protection. I know! Here come with me!

You first need to create 3 POKEMOZ and allow the player to choose which he wants. These 3 POKEMOZ are:

- Bulbasoz (grass)
- Oztirtle (water)
- Charmandoz (fire)

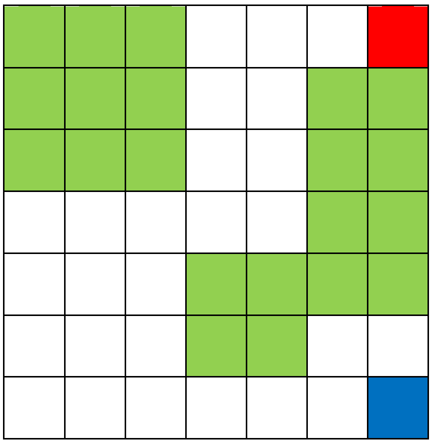
1.1. Map

Maps are represented as a matrix built with a tuple of tuples. This matrix contains either values **0 or 1**. A value of **1** represents the fact that there is tall grass in this position. A value of **0** represents the fact that a road passes through this position.

What follows is an example of a small map (the *r* stands for *row*). The starting point is **0** is at the very bottom right, while the goal is to reach the **0** at the very upper right.

The map will be stored in a text file (*.TXT). Map.txt will be the default map (the example above), MapX.txt are custom map that you can create. The default size is a grid of 7x7 cells but you can create other size of maps. Only the default map is needed for the evaluation of the project.

```
map (r(1 1 1 0 0 0 0)
     r(1 1 1 0 0 1 1)
     r(1 1 1 0 0 1 1)
     r(0 0 0 0 0 1 1)
     r(0 0 0 1 1 1 1)
     r(0 0 0 1 1 0 0)
     r(0 0 0 0 0 0 0))
```



1.2. POKEMOZ

POKEMOZ P are characterized by their type (grass, fire or water), their name (either beginning or ending with “oz”), their health point (remaining/maximum HP), their level (LX), and their experience (XP).

Their type should be represented either with the text (grass/fire/water), or with a colored box (green for grass, orange or red for fire, blue for water), or with a colored image.

Grass	🟩
Fire	🟥
Water	🟦

POKEMOZ can only live in tall grass or be carried by a POKEMOZ trainer. Wild POKEMOZ cannot be found outside tall grass. You need to add a parameter ***Probability*** for the probability to find a wild POKEMOZ in tall grass, this parameter must be in the range [0, 1] meaning from 0% to 100% probability to find a wild POKEMOZ. This is an external parameter that will be set at the start of the game.

1.3. Pokemoz trainers

POKEMOZ trainers T have one and only one POKEMOZ (bonus: you can try to support more if you want).

They can either walk on roads or on tall grass. Your POKEMOZ trainer can either be moved manually (using the arrow keys) or automatically at a certain speed. Other POKEMOZ trainers are spread over the map and can move freely. You decide how and when other POKEMOZ trainers move. The speed of your POKEMOZ trainer must be a parameter ***Speed*** that will range [0, 10] where the real speed will be processed from the formula **((10 - Speed) * DELAY)**. DELAY is a constant that can be adjusted (e.g., by default you can use 200ms).

1.4. Fights

There are two types of fights: a trainer against another one, or a trainer against a wild POKEMOZ.

1.4.1. Trainer VS Wild POKEMOZ

In case your POKEMOZ trainer meets a wild POKEMOZ in the tall grass, they can either fight or run away. Each time your POKEMOZ trainer moves on tall grass, she/he can fall against a wild POKEMOZ.

In order to test the game, you should add the parameter ***AutoFight*** to set if your POKEMOZ trainer always fight, always run away or if you let the choice every time you fall against a wild POKEMOZ (bonus: You can also provide a special algorithm to choose if you want to fight or run away depending on the level and type of the POKEMOZ you are fighting against).

The attack can damage the POKEMOZ's health according to this table:

	🟩	🟥	🟦
🟩	2	1	3
🟥	3	2	1
🟦	1	3	2

This means that if a grass-typed POKEMOZ attacks another grass-typed POKEMOZ the attack will make 2 damage points, only 1 damage point if it attacks a fire-typed POKEMOZ and 3 damage points if it attacks water-typed POKEMOZ. This means that fire-typed are good against grass-typed POKEMOZ, grass-typed POKEMOZ are good against water-typed POKEMOZ, and water-typed POKEMOZ are good against fire-typed POKEMOZ. When a POKEMOZ got hit by the other POKEMOZ its HP are decreased by the damage points.

An attack can either succeed or fail. Here is a formula to define the probability (in %) for an attack to succeed **((6 + LA – LD) * 9)** where **LA** is the level of the attacker and **LD** is the level of the defender. So that the POKEMOZ with highest level gets higher probability of hitting the other POKEMOZ.

The fight is over when a POKEMOZ has no more HP remaining. A POKEMOZ without HP remaining is cannot be used anymore (bonus: you can add special place to rest your POKEMOZ if you want).

When one of your POKEMOZ wins a fight its experience is increased by the level of the wild POKEMOZ he was fighting against. If your POKEMOZ has 0 XP and wins a fight again a POKEMOZ at level 5, your POKEMOZ's experience is increased by 5 XP (for a total of 5 XP). If your POKEMOZ then wins against a level 6 POKEMOZ then its experience will be increased by 6 XP (for a total of 11 XP).

1.4.2. Trainer VS Trainer

When two POKEMOZ trainers are next to each other, they have to fight. The fight can either be simulated or done manually. A POKEMOZ trainer cannot run away from a fight.

The fight is over when one of the POKEMOZ trainers has no more POKEMOZ with life remaining.

1.5. Evolution

There is no game if there is no evolution. A POKEMOZ always starts at level 5 with 0 XP and a maximum of 20 HP. All the POKEMOZ starts the game with maximum health (20/20 HP). Wild POKEMOZ always starts a new fight with maximum health, but their level can increase with time (you are free to choose how and when to increase the level of wild POKEMOZ).

Here is a table of the different levels:

Level	*HP*	*XP needed*
5	20	-
6	22	5
7	24	12
8	26	20
9	28	30
10	30	50

When a POKEMOZ reaches a new level, the remaining HP is set to the new maximum. This means that your POKEMOZ starts at 20 HP, but after earning more than 5 XP, its maximum will be set to 22 HP and its health will be fully recovered (22/22 HP).

2. Simulation

Your project consist in modeling and implementing a complete simulation for the game described above. For this, you have to use the abstractions introduced during the course before the eastern holidays. You have to provide a graphical user interface that allows to appreciate the interaction between the different components. You can **only** use the declarative model abstractions extended with ports. It is not allowed to use things like *NewCell*, *IsDet*, *IsFree*, classes, or whatever language abstraction out of the declarative model and ports.

3. The game

The most important part of the project is to move your POKEMOZ trainer within the map from the starting point toward the goal. A possible way of designing your application goes as follows. In the lift control system (that will be seen in Lab10), every lift has a position in the building. In this project each POKEMOZ trainer has a position on the map. To make a move, your POKEMOZ trainer inspects its environment in the map, and takes a decision for its next step. Each decision is taken in order to move towards the goal while keeping your POKEMOZ with at least 1 HP remaining.

You should be really careful about the following cases:

- Two POKEMOZ trainers cannot be in the same place at the same time.
- No wild POKEMOZ should be found outside tall grass
- POKEMOZ fights are only 1 vs 1 at the same time (bonus: although you can change POKEMOZ during a fight if you allow a POKEMOZ trainer to carry more than one POKEMOZ)

Note that these are only some of the cases you should be aware of. In general, the idea is that the game is coherent, does not have race conditions on the shared state, and are funny to look at, or to play.

4. Bonus

There are several bonuses across the project's statement. However it does not mean you gain 1 point per bonus. The bonuses can help the game to be funnier to play and will positively play for your project's point but if the game itself does not work, bonuses will not be that useful.

5. Deliverables

The project has to be done in groups of two. We strongly discourage you to work alone because of the load of the project, and also because working with somebody else will help you to see things from a different point of view, and, it will develop your social skills too.

There are two deliverables for this project:

1. Designing your project (5 pts). You will have to submit a report of maximum 3 pages. The report will describe the architecture of your system presented in three parts: component diagram, state diagrams, and a discussion of data structures you plan to use. There is no much text on the report, but it must be in English.
2. The project running (15 pts). You have to submit a functor (or a set of functors) containing the program. The program receives several arguments as parameters: a text file containing the map, the value of ***Probability***, the value of ***Speed***, and the value of ***Autofight***. You must provide a README file with the instructions on how to compile and run the game. Makefiles are welcome but not mandatory.
3. The deadline: Deadline: Monday, May 4, 11:58 PM (23h58).
4. An appointment will be scheduled on May 5 and May 6 for you to defend and show your project to the assistant.

The document justifying your design must be in PDF format. The README file can be a raw text file. The whole project code (no OZF files allowed) must be zipped in a ZIP file with the last name of the members (e.g., melchior-melchior.zip).

QTk will be used for the graphical user interface (GUI). You can find more information about QTk and other resources on iCampus (http://www.icampus.ucl.ac.be/).

How to submit: the PDF and ZIP files must be sent by e-mail at jeremie.melchior@uclouvain.be. Please use “[LINGI1131 – Pokemoz submission] lastname1 – lastname2” as subject (replace lastname by the last name of the people in the group).