



Advanced Programming, Spring 2023-2024

by Lect. Malek Malkawi

due by Monday, May 13, 2024 - 23:59

Mini Project 2

(Please carefully review the submission guidelines on the final pages before you begin, and ensure you understand all the requirements)

In modern era, managing your project according to customer expectation is crucial. Nowadays whatever you do is recorded, transformed to a form of data and stored or processed to benefit from you better. In this mini project, you are going to create your own Pokemon game and collect the data from players; then store, process and analyze it. Does that sound ethical? Well, it doesn't matter in modern world!

How should it look like?

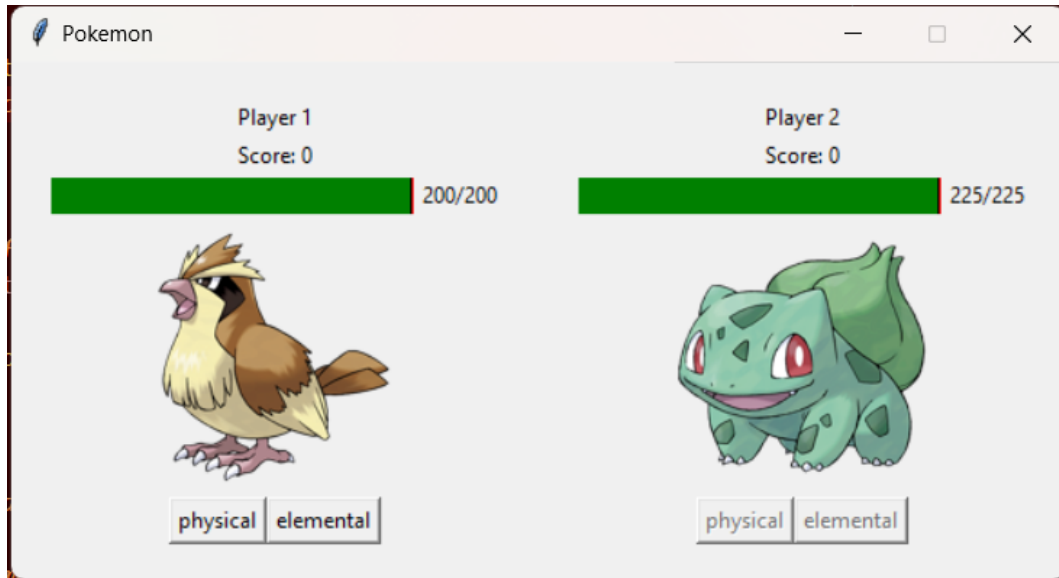
Graphical user interface is a bit detailed in this project, it is going to consist of more than one layout. So let's advance step by step.

How should it work:

In our Pokemon game, main game flow consists of the battle of two players. If you have ever played any Pokemon game before, it will seem pretty familiar, of course a bit simplified.

In Pokemon games, two Pokemons battle at once, and battles are performed as turn based, which means one player moves first then it becomes the other player's turn to move, just like in chess.

The picture shows the war screen, which is the scene that battle between two Pokemons



will occur. Two players will choose their Pokemons and after selection player 1 will move first. Players can choose two different attack types, first one is physical, which is the more consistent one, second one is elemental, which is the less consistent but high risk - high reward type of move.

Each Pokemon will have it's own element type and this element type will matter in battles if players prefer using elemental attacks. Fire is stronger than nature and water is stronger than fire etc.

But before going this far, let's start from the very beginning, from getting the data of the Pokemons.

1. If you are familiar with the Pokemon culture, there are hundreds of Pokemons in the Pokemon universe. Also every Pokemon has an element, a sub element, generation, some of them are legendary etc. so as you can see there are too many details for a simple child play. In this project, we are going to simplify things a lot and use only the data we needed. So first, see the figure below, which is the **Pokemon.csv** file that we are going to give you to do some data cleaning on it, we are going to need only 18 of them for this game.

In our game, rules are simple: two Pokemons fight, one that finishes enemy's health faster wins. So Pokemons are going to have only those attributes: Name, Element type, HP and attack values. From the given Pokemon database, **you will drop all**

```

1 #,N C:\poken\Abol\Programat\Software Projects\W02\pokemon.csv ense,Sp, Atk,Sp. Def,Speed,Generation,Legendary
2 1,Bulbasaur,Grass,Poison,318,45,49,49,65,65,45,1,False
3 2,Ivysaur,Grass,Poison,405,60,62,63,80,80,60,1,False
4 3,Venusaur,Grass,Poison,525,80,82,83,100,100,80,1,False
5 3,VenusaurMega Venusaur,Grass,Poison,625,80,100,123,122,120,80,1,False
6 4,Charmander,Fire,309,39,52,43,60,50,65,1,False
7 5,Charmeleon,Fire,405,58,64,58,80,65,80,1,False
8 6,Charizard,Fire,Flying,534,78,84,78,109,85,100,1,False
9 6,CharizardMega Charizard X,Fire,Dragon,634,78,130,111,130,85,100,1,False
10 6,CharizardMega Charizard Y,Fire,Flying,634,78,104,78,159,115,100,1,False
11 7,Squirtle,Water,314,44,48,65,50,64,43,1,False
12 8,Wartortle,Water,405,59,63,80,65,80,58,1,False
13 9,Blastoise,Water,530,79,83,100,85,105,78,1,False
14 9,BlastoiseMega Blastoise,Water,630,79,103,120,135,115,78,1,False
15 10,Caterpie,Bug,195,45,30,35,20,20,45,1,False
16 11,Metapod,Bug,205,60,20,55,25,25,30,1,False
17 12,Butterfree,Bug,Flying,395,60,45,50,90,80,70,1,False
18 13,Weedle,Bug,Poison,195,40,35,30,20,20,50,1,False
19 14,Kakuna,Bug,Poison,205,45,25,50,25,25,35,1,False
20 15,Beedrill,Bug,Poison,395,65,90,40,45,80,75,1,False
21 15,BeedrillMega Beedrill,Bug,Poison,495,65,150,40,15,80,145,1,False
22 16,Pidgey,Normal,Flying,251,40,45,40,35,35,56,1,False
23 17,Pidgeotto,Normal,Flying,349,63,60,55,50,50,71,1,False
24 18,Pidgeot,Normal,Flying,479,83,80,75,70,70,101,1,False
25 18,PidgeotMega Pidgeot,Normal,Flying,579,83,80,80,135,80,121,1,False
26 19,Rattata,Normal,253,30,56,35,25,35,72,1,False
27 20,Raticate,Normal,413,55,81,60,50,70,97,1,False
28 21,Spearow,Normal,Flying,262,40,60,30,31,31,70,1,False
29 22,Fearow,Normal,Flying,442,65,90,65,61,61,100,1,False
30 23,Ekans,Poison,288,35,60,44,40,54,55,1,False
31 24,Arbok,Poison,438,60,85,69,65,79,80,1,False
32 25,Pikachu,Electric,320,35,55,40,50,50,90,1,False
33 26,Raichu,Electric,485,60,90,55,90,80,110,1,False
34 27,SandsShrew,Ground,300,50,75,85,20,30,40,1,False
35 28,Sandslash,Ground,450,75,100,110,45,55,65,1,False
36 29,Nidoran♀,Poison,275,55,47,52,40,40,41,1,False

```

other columns.

Also the 18 Pokemons we are going to use are given below, **drop all others**:

- Bulbasaur
- Ivysaur
- Venusaur
- Charmander
- Charmeleon
- Charizard
- Squirtle
- Wartortle
- Blastoise
- Caterpie
- Metapod
- Butterfree

- Weedle
- Kakuna
- Beedrill
- Pidgey
- Pidgeotto
- Pidgeot

After this data cleaning done, your new and clean dataset should seem like this:

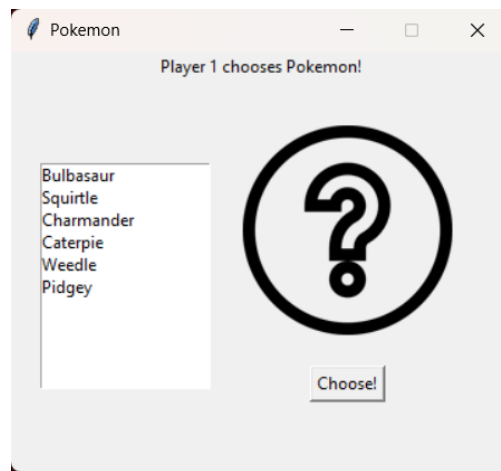
```
cleanPokemons.csv X
cleanPokemons.csv > data
1 ,Name,Element,HP,Attack
2 0,Bulbasaur,Grass,45,49
3 1,Ivysaur,Grass,60,62
4 2,Venusaur,Grass,80,82
5 3,Charmander,Fire,39,52
6 4,Charmeleon,Fire,58,64
7 5,Charizard,Fire,78,84
8 6,Squirtle,Water,44,48
9 7,Wartortle,Water,59,63
10 8,Blastoise,Water,79,83
11 9,Caterpie,Bug,45,30
12 10,Metapod,Bug,50,20
13 11,Butterfree,Bug,60,45
14 12,Weedle,Bug,40,35
15 13,Kakuna,Bug,45,25
16 14,Beedrill,Bug,65,90
17 15,Pidgey,Normal,40,45
18 16,Pidgeotto,Normal,63,60
19 17,Pidgeot,Normal,83,80
20
```

Use pandas for this task and write the clean database to a csv file.

You are going to use these Pokemon data in game, when players choose their Pokemons, their attributes will be taken from this file.

2. After getting the clean dataset, now we are ready to build the GUI for the game flow.

First, players need to choose their Pokemons, so we are going to need a Pokemon selection screen. Here you can build your own beautiful GUIs but it doesn't need to be too much detailed, or specifically the same with the selection screen system below, just create something that will do the task, which is just the selection of Pokemons.



Above you see the selection window for Player 1, player 1 will go to choose the name of Pokemon from the Listbox and then see the image of the Pokemon in the question marked place.

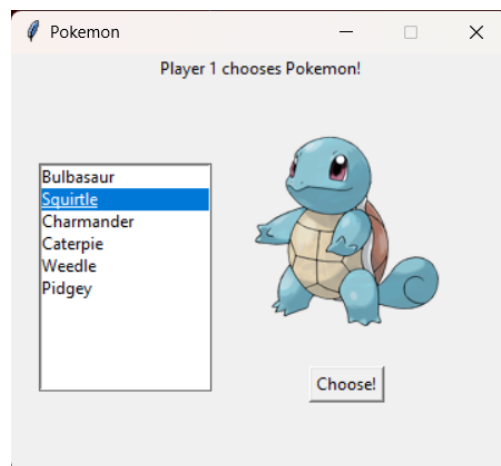
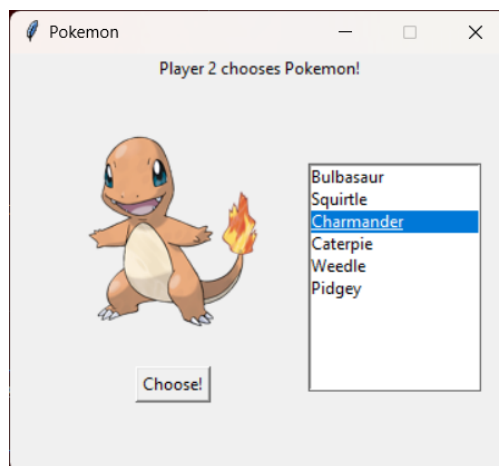
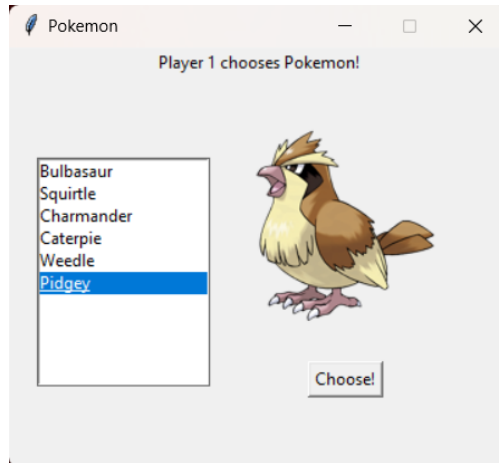


Image will change when player click any of the Pokemons in the Listbox. When player 1 decides the Pokemon he wants, then needs to click **Choose** button and then selection window for player 2 will appear.

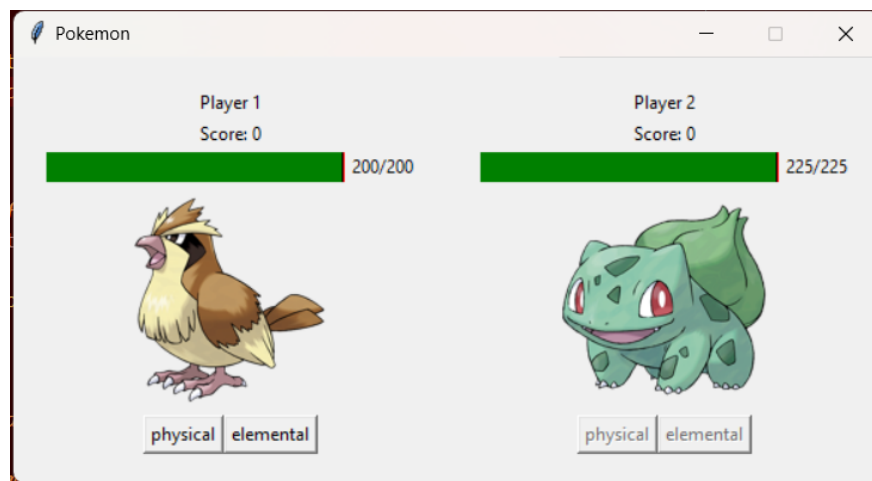


Above figures you see only 6 Pokemons for each player, but cleaned dataset had 18 of them? Where are the rest 12 Pokemons?

If you are not familiar with Pokemon universe, pokemons can evolve as they grow up or fight and get stronger. In the dataset we cleaned actually has 6 different Pokemons, and every Pokemons has their evolution level 2 and 3. In the battles, the Pokemon who won the round will get evolved and both it's attributes and image will change. So other 12 Pokemon will appear in the next phases of the battles.

After both players choose their Pokemons, battle will begin.

3. Battles are performed turn based, one player hits and after him other player hits. So be sure when it's one player's turn, other player has to wait and can not attack.



In the figure as you see, since it's Player 1's turn (Player 1 starts the game) buttons of Player 2 are disabled.

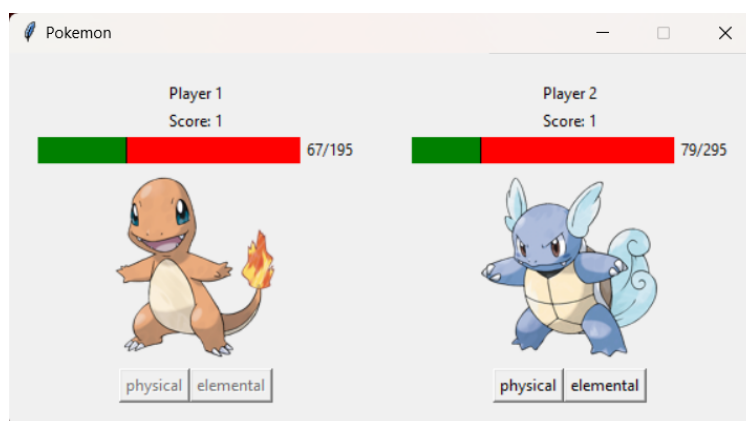
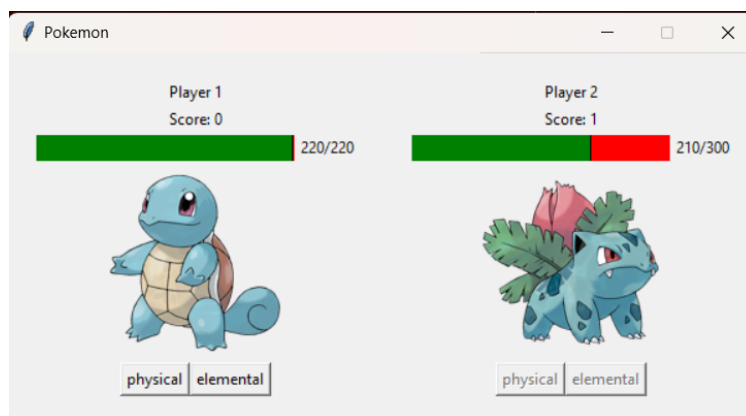
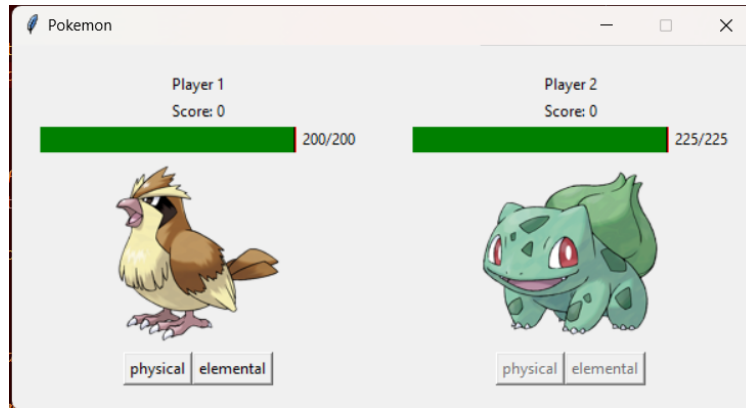
Players have two options: physical attack and elemental attack. Difference of physical and elemental attack is like as follows:

- **Physical:** Can give damage between 75%-100% of Pokemon's attack value.
- **Elemental:** Can give damage between 50%-100% of Pokemon's attack value, **if attacker is elementally stronger than defender Pokemon, damage will be doubled with a 80% chance.**

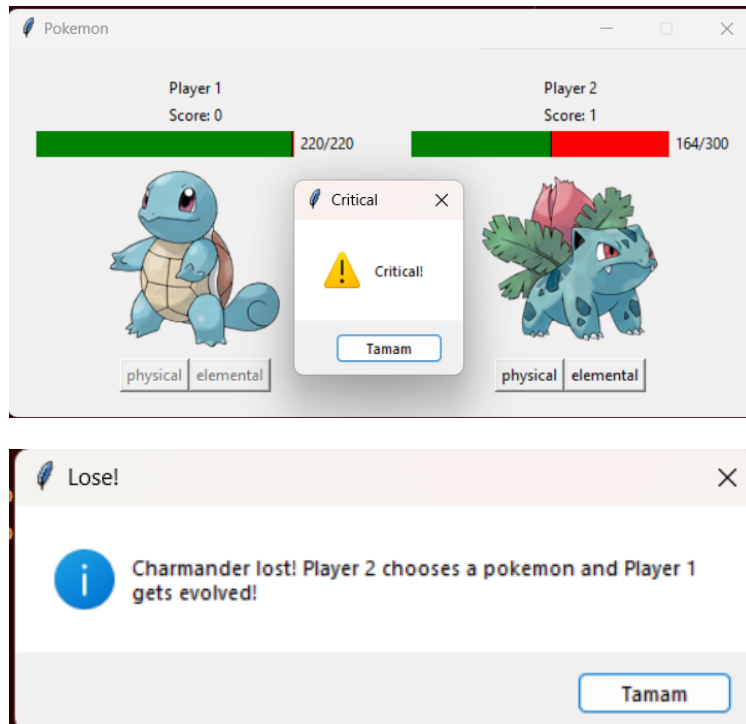
Here you need to add the functionality to buttons, they will do the calculations of attack

and decrease the HP of opponent, and visually represent it on the screen. Health bars of Pokemons will be updated after every attack.

See the sample screenshots of the battles below:

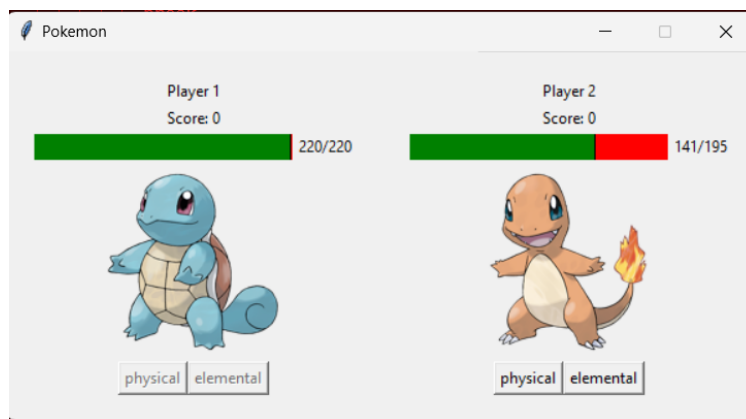


If any of player lose, an infobox should appear:

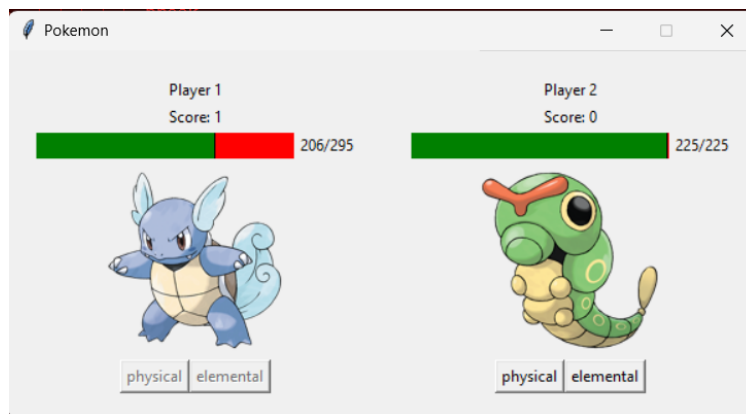


Here you need to implement evolution mechanic, loser will choose a pokemon again but winner will not choose a new pokemon, instead winner will have the same pokemon but evolved version of it. See the example of evolution below:

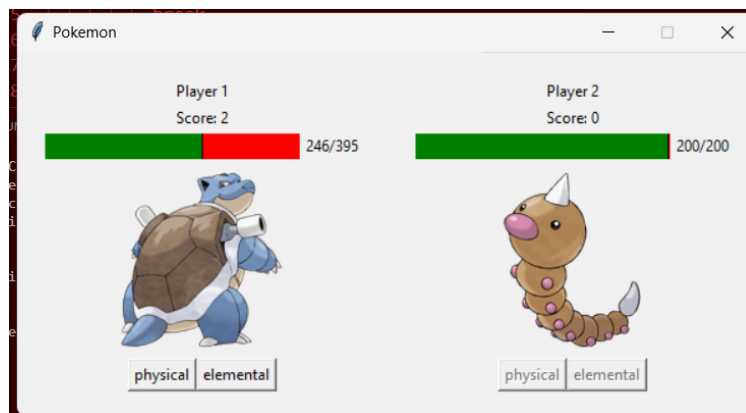
First Player 1 starts with Squirtle and Player 2 starts with Charmander:



Then Player 1 wins and gets evolved, since squirtle evolves to Wartortle, now he is Wartortle and player 2 chooses a new pokemon.



Finally in the round 3, since player 1 won again, he got his last evolution and Wartortle became a Blastoise and player 2 choose a new pokemon:

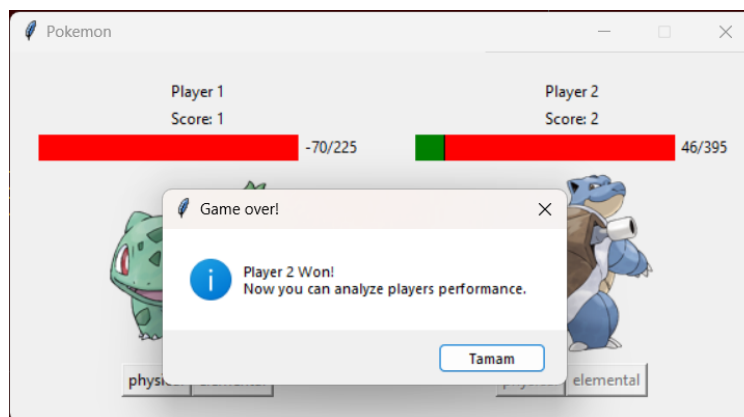
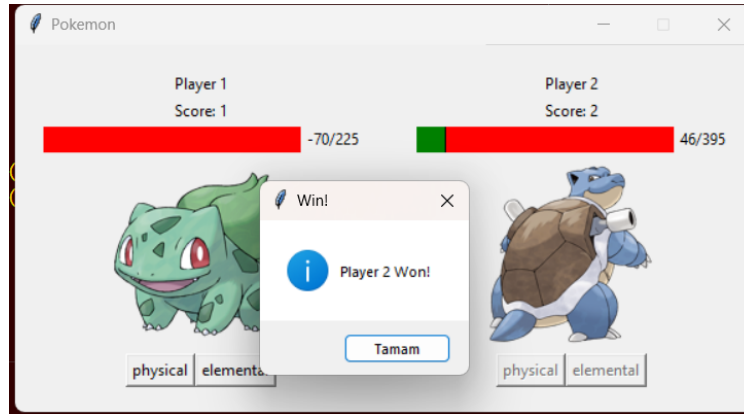


Also see player's scores, player 1's score increases every round but player 2's don't, when a player wins and gets evolved, his score increases. Who reaches to 3 first wins the whole game.

Here the logic is, winner gets evolved and gets stronger, but if he loses, he loses evolution and again chooses a level 1 pokemon. So if a player wants to be successful, he should be careful that he doesn't break the win streak, so gets evolved again and again, then win easily. Dont forget we only have 3 levels of pokemons; level 1 we choose from the listbox, if he wins once it gets level 2 and if wins again he gets level 3.

Before a new round begins, evolved player should revive with %70 of his max health, or it will be impossible to beat evolved pokemon.

When a player reaches score 3, game should end and an infobox should appear:



If game flow seemed confusing, definitely watch the video about the game flow, see how it should be implemented.

4. We assume battles are handled so far. In the last part, we are going to do data analysis. But instead of doing this on some random toy datasets, we are going to create our very own and do data analysis on it.

As Players play the game and do the battles, we are going to collect data from the game and see players' performances. We will create a dataframe with those data and then do data visualisation on it.

After every attack, collect those data and store in some array or list:

- Name of the attacker Pokemon
- Dealt damage value to opponent
- Remaining health value of opponent
- Was attack a critical hit or not (boolean data or 0's and 1's)
- Was attack an elemental attack or not (boolean data or 0's and 1's)

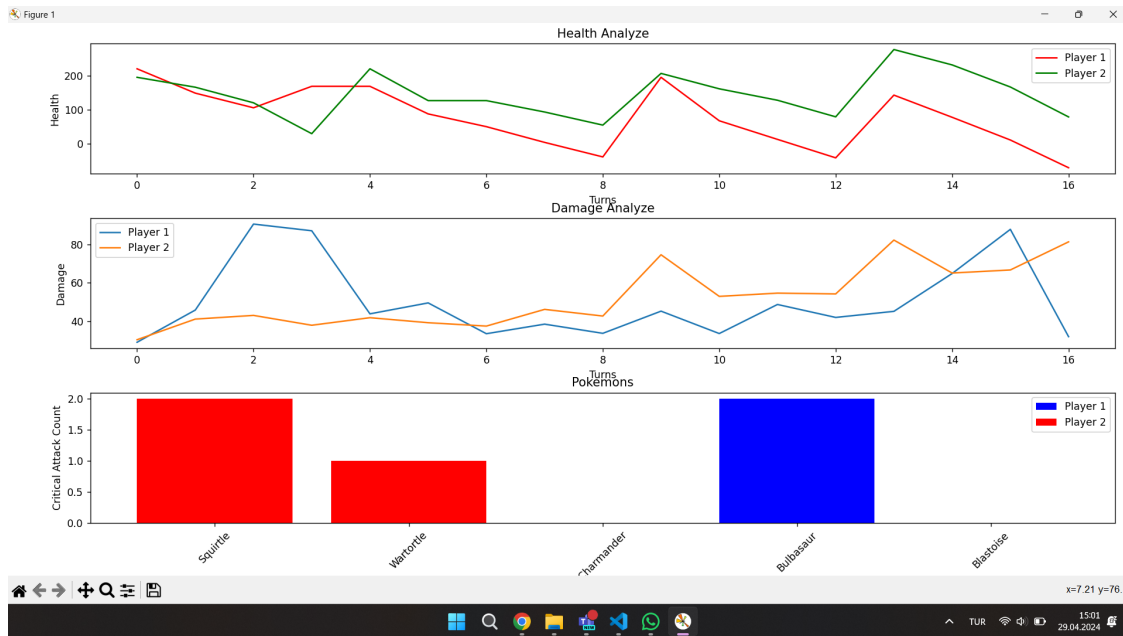
At the end of the battle, all those data needs to be collected. When battle ends, by using these data, create a dataframe and write it to an **excel file**. Final excel file should be something like this:

	A	B	C	D	E	F	G	H	I	J	K
1		Pokemon 1	Pokemon 2	Health 1	Health 2	Damage 1	Damage 2	Critical 1	Critical 2	Element 1	Element 2
2	0	Squirtle	Charmander	220.0	195.0	28.99850368354	30.32160948265	0.0	0.0	1.0	1.0
3	1	Squirtle	Charmander	148.6176524255	166.0014963164	45.76253186327	41.06073809179	0.0	0.0	1.0	0.0
4	2	Squirtle	Charmander	105.6343473252	120.2389644531	90.52655034382	42.98330510029	1.0	0.0	1.0	0.0
5	3	Squirtle	Squirtle	168.6477177815	29.71241410935	87.04518265972	37.85228221846	1.0	0.0	1.0	0.0
6	4	Wartortle	Squirtle	168.6477177815	220.0	43.79216589137	41.78893069241	0.0	0.0	1.0	1.0
7	5	Wartortle	Squirtle	87.68504798614	126.7193268554	49.48850725319	39.17373910297	0.0	0.0	0.0	0.0
8	6	Wartortle	Squirtle	50.26755859698	126.7193268554	33.44612371438	37.41748938915	0.0	0.0	1.0	0.0
9	7	Wartortle	Squirtle	4.151206547990	93.27320314105	38.42646543536	46.11635204899	0.0	0.0	1.0	0.0
10	8	Wartortle	Squirtle	-38.4950647607	54.84673770575	33.64766832552	42.64627130872	0.0	0.0	1.0	0.0
11	9	Charmander	Wartortle	195.0	206.5	45.19577582914	74.56217878394	0.0	1.0	1.0	1.0
12	10	Charmander	Wartortle	67.54987606222	161.3042241708	33.53561075536	52.88794515382	0.0	0.0	1.0	0.0
13	11	Charmander	Wartortle	12.97202861447	127.7686134154	48.66859183222	54.57784744775	0.0	0.0	1.0	0.0
14	12	Charmander	Wartortle	-41.1922235813	79.10002158326	41.92351803661	54.16425219579	0.0	0.0	1.0	0.0
15	13	Bulbasaur	Blastoise	142.8082289645	276.5	45.09457773676	82.19177103548	0.0	0.0	1.0	0.0
16	14	Bulbasaur	Blastoise	77.76695205464	231.4054222632	64.78431625511	65.04127690986	1.0	0.0	1.0	0.0
17	15	Bulbasaur	Blastoise	11.12604882875	166.6211060081	87.78321895745	66.64090322588	1.0	0.0	1.0	0.0
18	16	Bulbasaur	Blastoise	-70.1495813937	78.83788705072	31.91773123281	81.27563022254	0.0	0.0	1.0	0.0

Now we have our very own dataset, now it's time to do some data visualisation!

- Finally, in the last part, you are going to come up with the graphics of Damage, Health and Crit columns' data. Crit graph will be a bar graph and it will show every name of pokemon played in game and how many times did they hit a critical attack.

You can put all the plots in a single window or show them separately, an example of the graphs are given below:



Implementation Notes:

- We highly recommend you to use classes in this project, they can lower your burden a lot and make it easier to manage this comprehensive project. For example you can use a pokemon class, a gui class to manage tkinter and other classes that can make it easier to manage the code.
- Images in tkinter can be so annoying and hard to manage and use, so if you face problems when you deal with images, don't worry, you don't have to use images at all, but if you do, it will be appreciated.
- Apart from the above, the lecture slides are enough for most parts, still feel free to research and implement. Please provide references if you use outer sources. TutorialsPoint, StackOverflow and Effbot might be good starting points, but **do not copy codes directly**.

How and when do I submit my project? :

- Make sure your Python code compiles. Any non-compiling code will receive **very low mark**.

- Submit your own code in a single PYTHON (.py) file. Name your code file as the naming policy (see below).
- Naming : ID_YourDept_InitialName_LastName_MP2.py, for example, 6421345CoEMalek-MalkawiMP2.py (Do not use any Turkish characters in file name)
- **Late Submission Policy: -%10 for 2 hour, -%15 for 12 hours, -%20 for 24 hours, -%40 for 48 hours, more than 48 hours not accepted.**
- Those who **do not** follow the above naming conventions **will get 10 pts off** of their grade.

Warnings:

- You **CANNOT** use place for geometry, only grid and pack are allowed.
- Do not talk to your classmates on project topics when you are implementing your projects (This is serious). Do not show or email your code to others (This is even more serious). If you need help, talk to your TAs or the instructor, not to your classmates. If somebody asks you for help, explain to them the lecture slides, but do not explain any project-related topic or solution. Any similarity in your source codes will have serious consequences for both parties.
- Carefully read the project document, and pay special attention to sentences that involve “**should**”, “**should not**”, “**do not**”, and other underlined/bold font statements.
- If you use code from a resource (website, book, etc.), make sure that you reference those resources at the top of your source code file in the form of comments. You should give details of which part of your code is from what resource. Failing to do so **may result in** plagiarism investigation.

Grading Criteria:

Have further questions?: Please contact your TAs if you have further questions. If you

Functionality					
Create Pokemon dataset and write it to a csv file	Build GUI for the Pokemon dataset	Battle design and implemen- tation	Create battle details dataset and write it to a csv file	Create visualization for the battle details analysis	Others
15 pts	25 pts	25 pts	15 pts	15 pts	5 pts

need help with anything, please use the office hours of your TAs and the instructor to get help. **Do not walk in randomly (especially on the last day) into your TAs' or the instructor's offices. Make an appointment first. This is important. Your TAs have other responsibilities. Please respect their personal schedules!**

Good Luck!