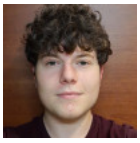
Imprimir

PSI Group C Wiki

Group members

Nicolás Sousa González



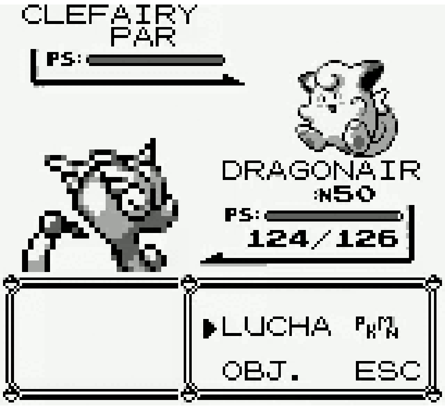
Alejandro Mateo Costa de Dios

Javier Bilbao Lima

Juan Pontón Rodríguez

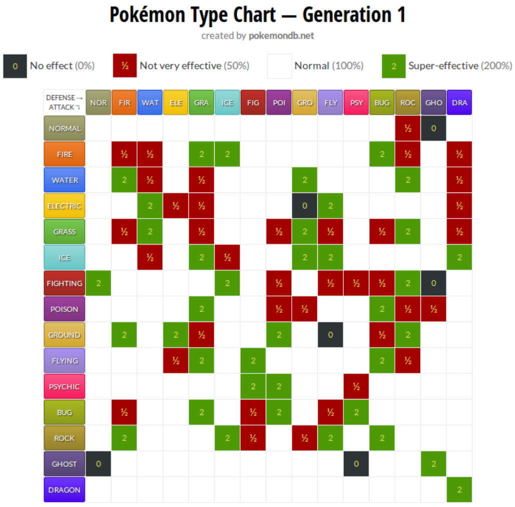
Project explanation

The project will involve the implementation of an AI capable of winning in the battle system of the first-generation Pokemon game. The Pokemon battle system consists of two opponents, each with the same number of Pokemon (fighting characters), which in our application will be three. The goal of the battle is to defeat all of the opponent's Pokemon. The battle is turn-based, in which you can choose a move for your Pokémon to use, or you have the option to switch to another Pokémon that is still alive.



Each Pokemon has specific and distinct characteristics from the others. Each Pokemon has four statistics: Attack, Defense, Special, and Speed, they have four possible moves and they can also be of one or two types.

Types: Pokemon can be of one or two types, and in the first generation, there are a total of 15 types. Each type is weak or strong against other types, depending on the type of Pokemon they are facing. When one type is strong against another, the damage dealt to the other Pokémon is doubled, and in case it's weak, the damage is halved. There's also the possibility that one type has no effect on another.



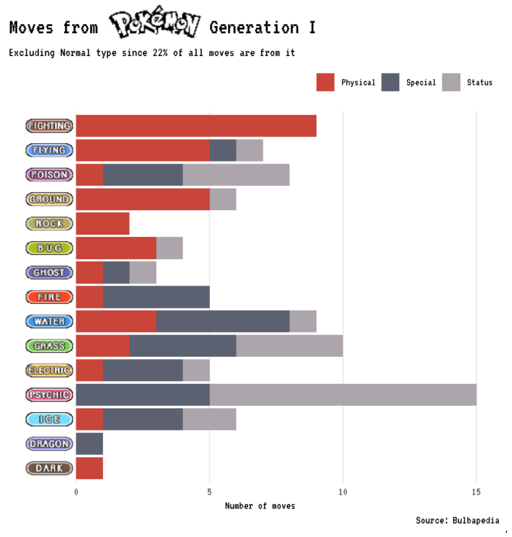
Pokémon Type Chart — Generation 1

created by pokemondb.net

0 No effect (0%) ½ Not very effective (50%) Normal (100%) 2 Super-effective (200%)

DEFENSE → ATTACK ↴	NOR	FIR	WAT	ELE	GRA	ICE	FIG	POI	GRO	FLY	PSY	BUG	ROC	GHO	DRA
NORMAL													½	0	
FIRE		½	½		2	2						2	½		½
WATER		2	½		½				2				2		½
ELECTRIC			2	½	½				0	2					½
GRASS		½	2		½			½	2	½		½	2		½
ICE			½		2	½			2	2					2
FIGHTING	2					2		½		½	½	½	2	0	
POISON					2			½	½			2	½	½	
GROUND		2		2	½			2		0		½	2		
FLYING				½	2		2					2	½		
PSYCHIC							2	2			½				
BUG		½			2		½	2		½	2				
ROCK		2				2	½		½	2		2			
GHOST	0										0			2	
DRAGON															2

Moves: Each Pokemon has a total of four moves, and these moves are of a specific type, for example, fire type. These move types are independent of the Pokemon's own type, meaning a water-type Pokemon could learn a fire-type move, although it's not common. Moves can be physical, special or Status(not implemented). Each move is associated with a power and an accuracy. The higher the power, the more damage it deals, and accuracy determines how likely the attack is to hit. These statistics can be displayed during the battle.



Statistics:

- Attack (Atk): Represents the strength of the Pokemon's physical moves.
- Defense (Def): Represents the Pokemon's ability to resist damage from the opponent's physical moves.
- Speed (Speed): Determines how quickly a Pokemon acts in battle. If a Pokemon has higher speed than the opponent, it will attack first, and if they have the same speed, it's resolved with a 50/50 chance.
- Special Attack (SpA): the power of special moves .
- Special Defend(SpD): the resistance to the opponent's special moves.
- Hit Points (HP): Represents the amount of damage a Pokemon can take before fainting, i.e., being defeated.

These statistics can be displayed during the battle, but only for the player's own Pokémon, and to give an advantage to inexperienced players, a range of possible speeds for the opponent's Pokémon will be provided to guide the user.

Knowing all the above, the damage caused when performing an attack on a Pokémon is calculated as follows:

$$Damage = \left(\frac{2 \times Level + 10}{250} \times \frac{Attack}{Defense} \times Base + 2 \right) \times Modifier$$

$$Modifier = STAB \times Type \times Critical \times other \times (random \in [0.85, 1])$$

Level: Same level for all Pokémon.

Attack: The attack stat of the Pokémon using the move.

Defense: The Defense stat of the enemy Pokémon.

Base: The power of the move used.

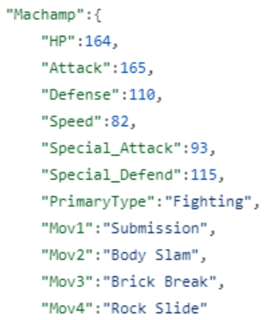
STAB: By default it is 1, but If the move used is the same type of the Pokémon, this value is 1.5.

Type: If the move type is strong versus the type of the opponent Pokémon this value is 2, if the move type is weak versus the type of the opponent Pokémon this value is 0.5 and if the move type is No effect versus the type of the opponent Pokémon this value is 0.

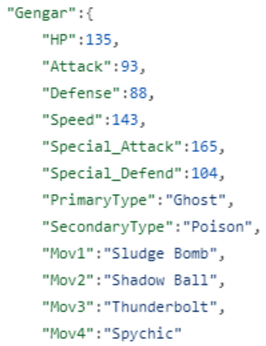
Critical: By default it is 1, but there is a 6.25% chance of a critical hit that would do double damage, in this case, the value of Critical is 2.

Other: This value will not be used.

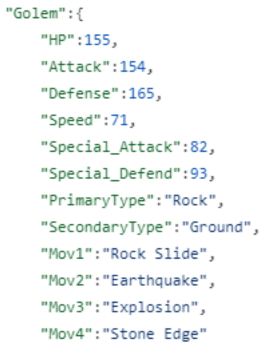
Set of Pokémons:



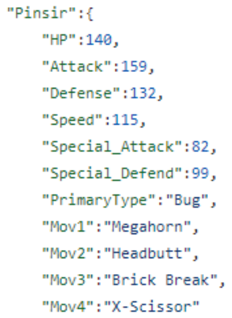
```
"Machamp":{
    "HP":164,
    "Attack":165,
    "Defense":110,
    "Speed":82,
    "Special_Attack":93,
    "Special_Defend":115,
    "PrimaryType":"Fighting",
    "Mov1":"Submission",
    "Mov2":"Body Slam",
    "Mov3":"Brick Break",
    "Mov4":"Rock Slide"
```



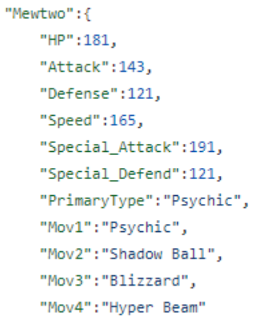
```
"Gengar":{
    "HP":135,
    "Attack":93,
    "Defense":88,
    "Speed":143,
    "Special_Attack":165,
    "Special_Defend":104,
    "PrimaryType":"Ghost",
    "SecondaryType":"Poison",
    "Mov1":"Sludge Bomb",
    "Mov2":"Shadow Ball",
    "Mov3":"Thunderbolt",
    "Mov4":"Spychic"
```



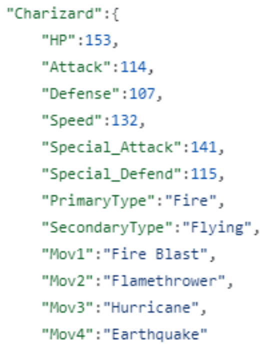
```
"Golem":{
    "HP":155,
    "Attack":154,
    "Defense":165,
    "Speed":71,
    "Special_Attack":82,
    "Special_Defend":93,
    "PrimaryType":"Rock",
    "SecondaryType":"Ground",
    "Mov1":"Rock Slide",
    "Mov2":"Earthquake",
    "Mov3":"Explosion",
    "Mov4":"Stone Edge"
```



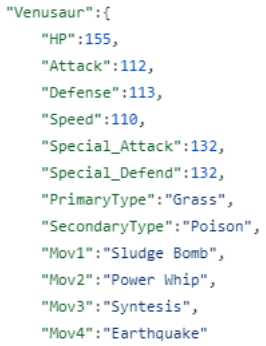
```
"Pinsir":{
    "HP":140,
    "Attack":159,
    "Defense":132,
    "Speed":115,
    "Special_Attack":82,
    "Special_Defend":99,
    "PrimaryType":"Bug",
    "Mov1":"Megahorn",
    "Mov2":"Headbutt",
    "Mov3":"Brick Break",
    "Mov4":"X-Scissor"
```



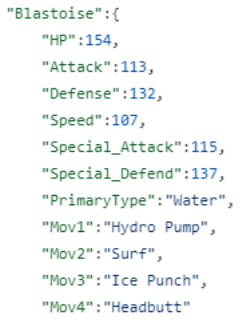
```
"Mewtwo":{
    "HP":181,
    "Attack":143,
    "Defense":121,
    "Speed":165,
    "Special_Attack":191,
    "Special_Defend":121,
    "PrimaryType":"Psychic",
    "Mov1":"Psychic",
    "Mov2":"Shadow Ball",
    "Mov3":"Blizzard",
    "Mov4":"Hyper Beam"
```



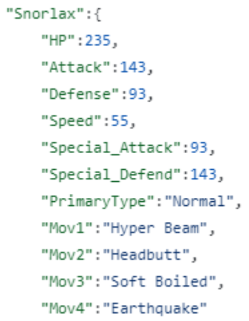
```
"Charizard":{
    "HP":153,
    "Attack":114,
    "Defense":107,
    "Speed":132,
    "Special_Attack":141,
    "Special_Defend":115,
    "PrimaryType":"Fire",
    "SecondaryType":"Flying",
    "Mov1":"Fire Blast",
    "Mov2":"Flamethrower",
    "Mov3":"Hurricane",
    "Mov4":"Earthquake"
```



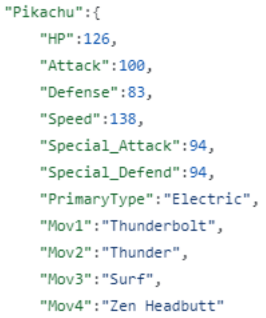
```
"Venusaur":{
    "HP":155,
    "Attack":112,
    "Defense":113,
    "Speed":110,
    "Special_Attack":132,
    "Special_Defend":132,
    "PrimaryType":"Grass",
    "SecondaryType":"Poison",
    "Mov1":"Sludge Bomb",
    "Mov2":"Power Whip",
    "Mov3":"Syntesis",
    "Mov4":"Earthquake"
```



```
"Blastoise":{
    "HP":154,
    "Attack":113,
    "Defense":132,
    "Speed":107,
    "Special_Attack":115,
    "Special_Defend":137,
    "PrimaryType":"Water",
    "Mov1":"Hydro Pump",
    "Mov2":"Surf",
    "Mov3":"Ice Punch",
    "Mov4":"Headbutt"
```

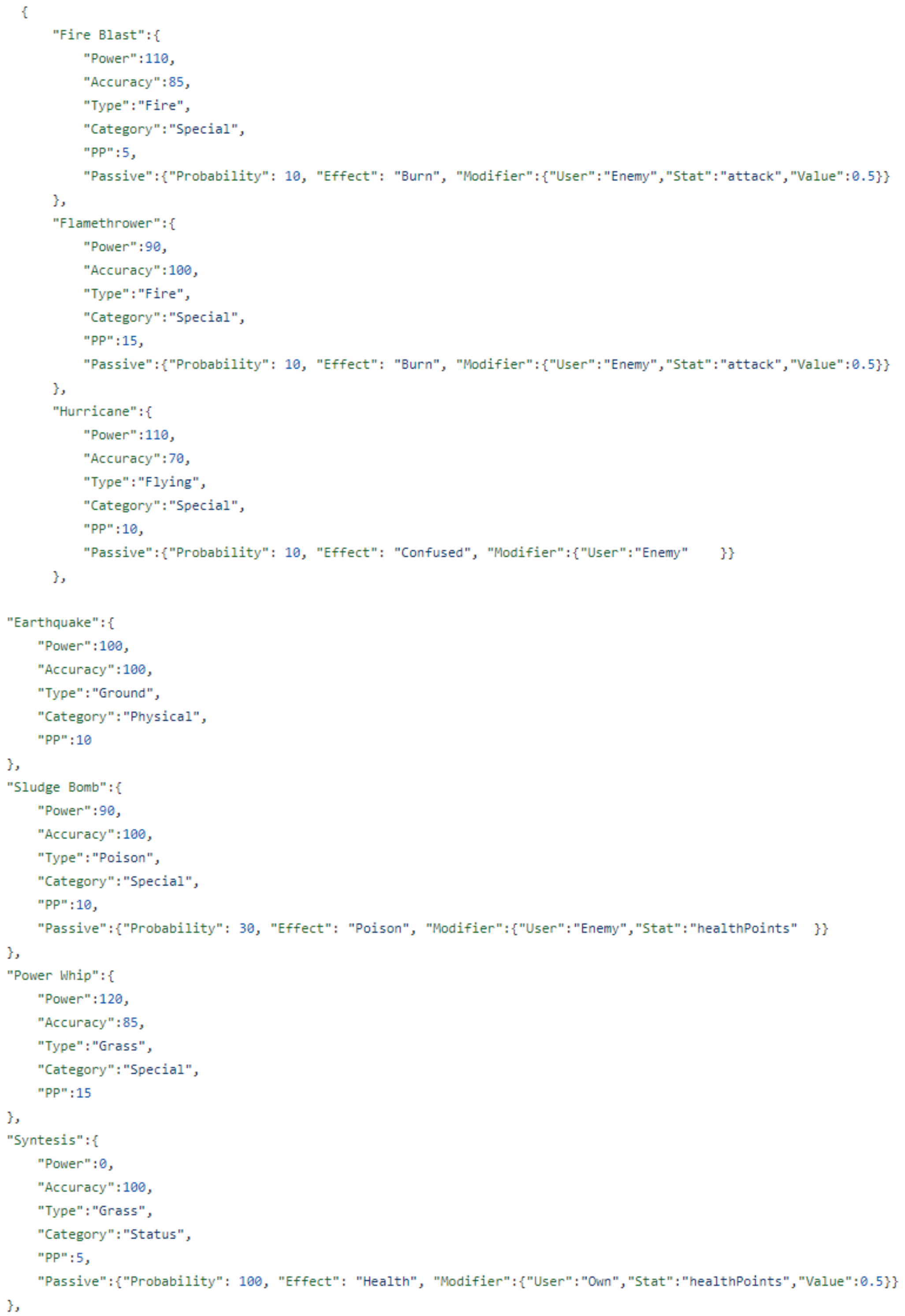


```
"Snorlax":{
    "HP":235,
    "Attack":143,
    "Defense":93,
    "Speed":55,
    "Special_Attack":93,
    "Special_Defend":143,
    "PrimaryType":"Normal",
    "Mov1":"Hyper Beam",
    "Mov2":"Headbutt",
    "Mov3":"Soft Boiled",
    "Mov4":"Earthquake"
```



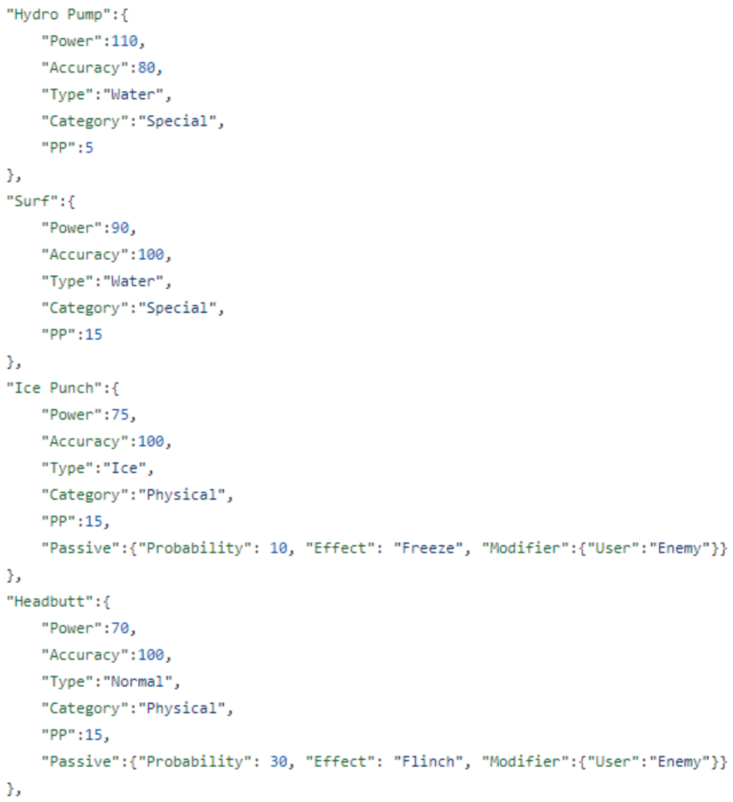
```
"Pikachu":{
    "HP":126,
    "Attack":100,
    "Defense":83,
    "Speed":138,
    "Special_Attack":94,
    "Special_Defend":94,
    "PrimaryType":"Electric",
    "Mov1":"Thunderbolt",
    "Mov2":"Thunder",
    "Mov3":"Surf",
    "Mov4":"Zen Headbutt"
```

Set of moves:

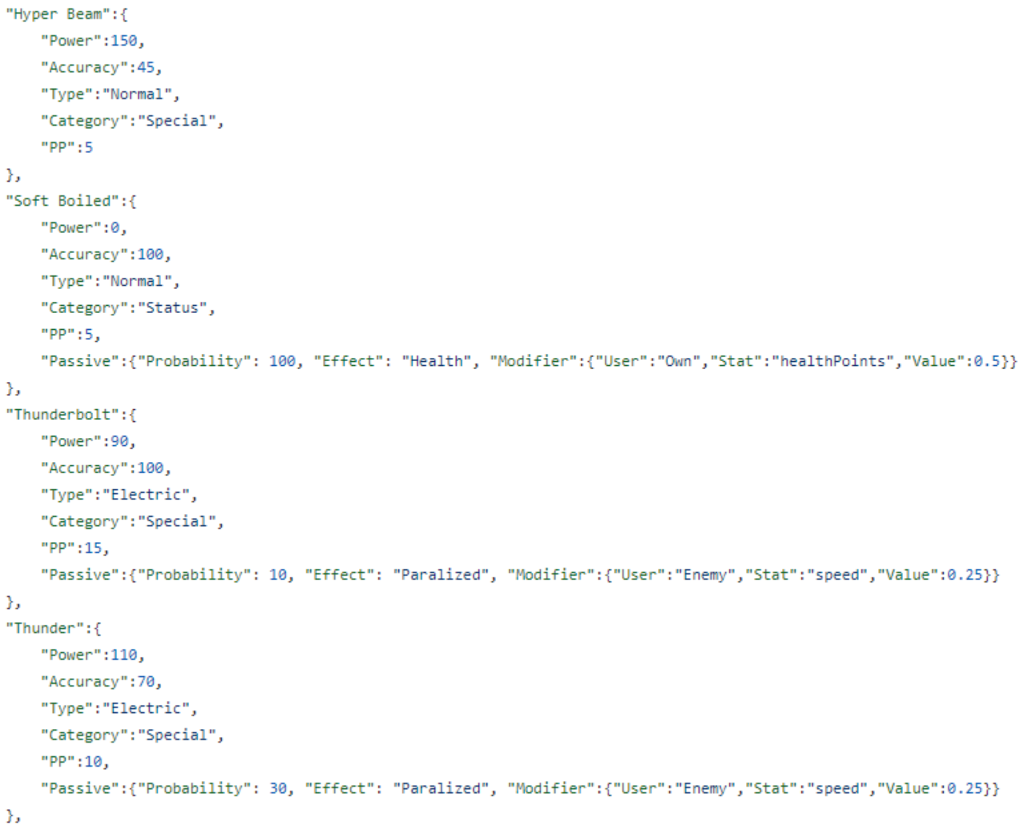


```
  {
      "Fire Blast":{
          "Power":110,
          "Accuracy":85,
          "Type":"Fire",
          "Category":"Special",
          "PP":5,
          "Passive":{"Probability": 10, "Effect": "Burn", "Modifier":{"User":"Enemy","Stat":"attack","Value":0.5}}
      },
      "Flamethrower":{
          "Power":90,
          "Accuracy":100,
          "Type":"Fire",
          "Category":"Special",
          "PP":15,
          "Passive":{"Probability": 10, "Effect": "Burn", "Modifier":{"User":"Enemy","Stat":"attack","Value":0.5}}
      },
      "Hurricane":{
          "Power":110,
          "Accuracy":70,
          "Type":"Flying",
          "Category":"Special",
          "PP":10,
          "Passive":{"Probability": 10, "Effect": "Confused", "Modifier":{"User":"Enemy"    }}
      },

"Earthquake":{
    "Power":100,
    "Accuracy":100,
    "Type":"Ground",
    "Category":"Physical",
    "PP":10
},
"Sludge Bomb":{
    "Power":90,
    "Accuracy":100,
    "Type":"Poison",
    "Category":"Special",
    "PP":10,
    "Passive":{"Probability": 30, "Effect": "Poison", "Modifier":{"User":"Enemy","Stat":"healthPoints"  }}
},
"Power Whip":{
    "Power":120,
    "Accuracy":85,
    "Type":"Grass",
    "Category":"Special",
    "PP":15
},
"Syntesis":{
    "Power":0,
    "Accuracy":100,
    "Type":"Grass",
    "Category":"Status",
    "PP":5,
    "Passive":{"Probability": 100, "Effect": "Health", "Modifier":{"User":"Own","Stat":"healthPoints","Value":0.5}}
},
```

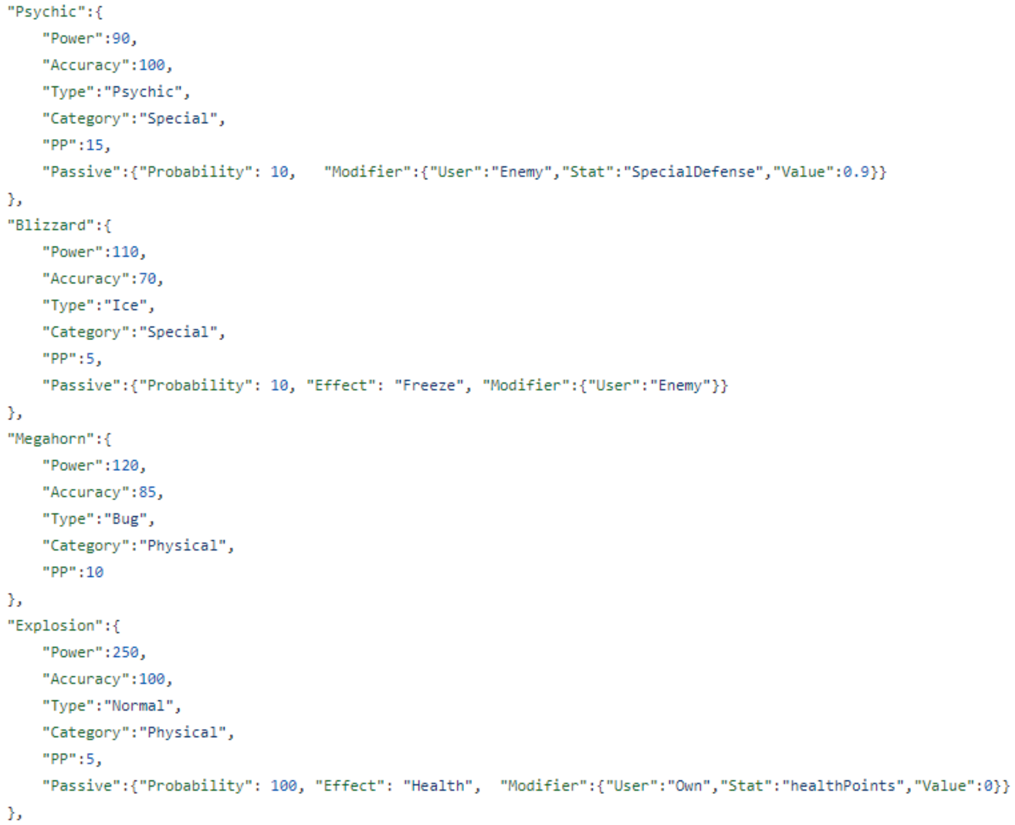


```
"Hydro Pump":{
    "Power":110,
    "Accuracy":80,
    "Type":"Water",
    "Category":"Special",
    "PP":5
},
"Surf":{
    "Power":90,
    "Accuracy":100,
    "Type":"Water",
    "Category":"Special",
    "PP":15
},
"Ice Punch":{
    "Power":75,
    "Accuracy":100,
    "Type":"Ice",
    "Category":"Physical",
    "PP":15,
    "Passive":{"Probability": 10, "Effect": "Freeze", "Modifier":{"User":"Enemy"}}
},
"Headbutt":{
    "Power":70,
    "Accuracy":100,
    "Type":"Normal",
    "Category":"Physical",
    "PP":15,
    "Passive":{"Probability": 30, "Effect": "Flinch", "Modifier":{"User":"Enemy"}}
},
```

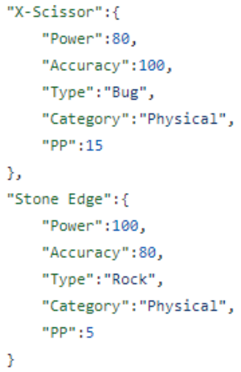


```
"Hyper Beam":{
    "Power":150,
    "Accuracy":45,
    "Type":"Normal",
    "Category":"Special",
    "PP":5
},
"Soft Boiled":{
    "Power":0,
    "Accuracy":100,
    "Type":"Normal",
    "Category":"Status",
    "PP":5,
    "Passive":{"Probability": 100, "Effect": "Health", "Modifier":{"User":"Own","Stat":"healthPoints","Value":0.5}}
},
"Thunderbolt":{
    "Power":90,
    "Accuracy":100,
    "Type":"Electric",
    "Category":"Special",
    "PP":15,
    "Passive":{"Probability": 10, "Effect": "Paralized", "Modifier":{"User":"Enemy","Stat":"speed","Value":0.25}}
},
"Thunder":{
    "Power":110,
    "Accuracy":70,
    "Type":"Electric",
    "Category":"Special",
    "PP":10,
    "Passive":{"Probability": 30, "Effect": "Paralized", "Modifier":{"User":"Enemy","Stat":"speed","Value":0.25}}
},
"Zen Headbutt":{
    "Power":80,
    "Accuracy":90,
    "Type":"Psychic",
    "Category":"Physical",
    "PP":15,
    "Passive":{"Probability": 20, "Effect": "Flinch", "Modifier":{"User":"Enemy"}}
},
"Submission":{
    "Power":80,
    "Accuracy":80,
    "Type":"Fighting",
    "Category":"Physical",
    "PP":15,
    "Passive":{"Probability": 20, "Effect": "Flinch", "Modifier":{"User":"Enemy"}}
},
"Rock Slide":{
    "Power":75,
    "Accuracy":90,
    "Type":"Rock",
    "Category":"Physical",
    "PP":15,
    "Passive":{"Probability": 30, "Effect": "Flinch", "Modifier":{"User":"Enemy"}}
},

"Brick Break":{
    "Power":75,
    "Accuracy":100,
    "Type":"Fighting",
    "Category":"Physical",
    "PP":15
},
"Body Slam":{
    "Power":85,
    "Accuracy":100,
    "Type":"Normal",
    "Category":"Physical",
    "PP":15,
    "Passive":{"Probability": 30, "Effect": "Paralized", "Modifier":{"User":"Enemy","Stat":"speed","Value":0.25}}
},
"Shadow Ball":{
    "Power":80,
    "Accuracy":100,
    "Type":"Ghost",
    "Category":"Special",
    "PP":15,
    "Passive":{"Probability": 20,   "Modifier":{"User":"Enemy","Stat":"SpecialDefense","Value":0.8}}
},
```



```
"Psychic":{
    "Power":90,
    "Accuracy":100,
    "Type":"Psychic",
    "Category":"Special",
    "PP":15,
    "Passive":{"Probability": 10,   "Modifier":{"User":"Enemy","Stat":"SpecialDefense","Value":0.9}}
},
"Blizzard":{
    "Power":110,
    "Accuracy":70,
    "Type":"Ice",
    "Category":"Special",
    "PP":5,
    "Passive":{"Probability": 10, "Effect": "Freeze", "Modifier":{"User":"Enemy"}}
},
"Megahorn":{
    "Power":120,
    "Accuracy":85,
    "Type":"Bug",
    "Category":"Physical",
    "PP":10
},
"Explosion":{
    "Power":250,
    "Accuracy":100,
    "Type":"Normal",
    "Category":"Physical",
    "PP":5,
    "Passive":{"Probability": 100, "Effect": "Health",  "Modifier":{"User":"Own","Stat":"healthPoints","Value":0}}
},
```



```
"X-Scissor":{
    "Power":80,
    "Accuracy":100,
    "Type":"Bug",
    "Category":"Physical",
    "PP":15
},
"Stone Edge":{
    "Power":100,
    "Accuracy":80,
    "Type":"Rock",
    "Category":"Physical",
    "PP":5
}
```

State of the art

Collection of similar works or apps with appropriate references.
Differences between the existing apps and yours.

1) The most similar work in comparison to our project is called Pokémon Showdown, which is an online battle simulator where you can challenge other people around the world from your browser, without the need of downloading anything.

Showdown! (pokemonshowdown.com)

Differences:

-Showdown is updated with every new generation of the franchise, so it contains the whole set of Pokémons, attacks, abilities, objects... In our project, we will reduce this aspects to a limited set from the first generation, in order to enable the correct development of the app before the established deadline.

-Pokémon Showdown, as we said earlier, is an online battle simulator where you compete against other players, but in our app users will face an AI trained to defeat them.

-You can play Showdown from any device only using your browser. On the other hand, our app will only be available on Android.

2) There multiple works from different people who use reinforcement learning to make an AI play the original Pokémon games, but we are going to take as reference one from Peter Whidden, as it is well documented and explained on his Youtube channel.

The main difference of our project with these type of works is that we do not have to worry about things like map exploration, reward collecting experience and all the topics that do not concern to the competitive combat.

Related work:

- AI Agent for Pokémon Battles
- Optimal Battle Strategy in Pokemon using Reinforcement Learning
- Gotta Train 'Em All: Learning to Play Pokémon Shodown with Reinforcement Learning
- Comparison of Artificial Intelligence Algorithms for Pokémon Battles
- Reinforcement Learning for a Turn-Based Small Scale Attrition Game
- Implementation and Evaluation of Information Set Monte Carlo Tree Search for Pokémon
- Learning complex games through self play - Pokémon battles
- Implementation and Evaluation of Information Set Monte Carlo Tree Search for Pokémon
- Combining ISMCTS with prediction of human decisions:

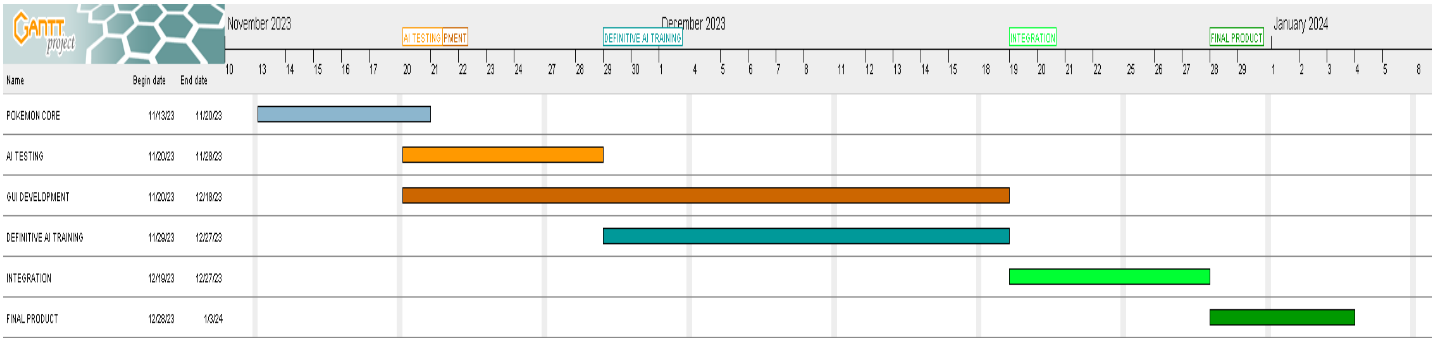
Chosing Monte Carlo Tree Search

The choice of Monte Carlo Tree Search (MCTS) as the foundation for the artificial intelligence in our Pokémon game simulator has been based on the evaluation of various strategies. Unlike other methods, such as Q-learning and neural networks, which, according to previous studies, yield mediocre results even with intensive training, MCTS emerges as a more robust option.The adaptability of MCTS stands out in the context of Pokémon. In a turn-based game, where information about the opponent's moves is not always known, MCTS effectively manages sequential decision-making, making it suitable for the dynamics of Pokémon battles.

In terms of Pokémon battles, characterized by the complexity arising from various species, moves, and strategies, MCTS addresses this intricate dynamic by exploring practically all possibilities, depending on the depth of exploration allowed. This approach enables it to focus on the most promising options through repeated simulations, overcoming the inherent complexity of the game.It is crucial to highlight that MCTS does not require intensive pre-training, unlike other strategies. Its ability to practically consider all possibilities, adjusting the depth of exploration, positions it as an approach that efficiently adapts to the changing complexities of the game.

Finally, it is worth noting the ease of implementation of MCTS for our game. Its straightforward structure and the ability to adjust parameters facilitate integration, allowing it to adapt specifically to the needs of our Pokémon simulator.

Project planning



Name	Begin date	End date
POKEMON CORE	11/13/23	11/20/23
AI TESTING	11/20/23	11/28/23
GUI DEVELOPMENT	11/20/23	12/18/23
DEFINITIVE AI TRAINING	11/29/23	12/27/23
INTEGRATION	12/19/23	12/27/23
FINAL PRODUCT	12/28/23	1/3/24

POKEMON CORE: Develop of the framework from which we're going to create the application, i.e. Pokemons class, etc. (It will be done by the four of us)

AI TESTING: Test different AI algorithms and see which could be implemented. (Javier and Juan)

GUI: Graphical user interface, i.e. android app(Nicolás and Alejandro Mateo)

DEFINITIVE AI TRAINING: One the final algorithm is chosen, approach the training from different sides to see which works better. (Javier and Juan)

INTEGRATION: Once the GUI and the AI is complete, join both into the app and debug. (It will be done by the four of us)

FINAL PRODUCT: Test the final version of the app in order to assure that it works fine. (It will be done by the four of us)

Final timing

The pokemon core first version was developed from the 10th of November to the 21th of the same month. Some small features were added after that date.

Before the AI implementation, a wide period of time was invested in researching related literature so that later it would be easier to implement the algorithm, therefore,

the AI testing and training were merched into simple AI development because of the chosen strategy and it was developed after the week of 18 of december.

The GUI was developed at the same time and after that the integration of both and the final product was done with small reviews and changes until the last week.

Resources used

Resources needed for their development and use, like tutorials, external libraries, etc.

Some online resources and courses:

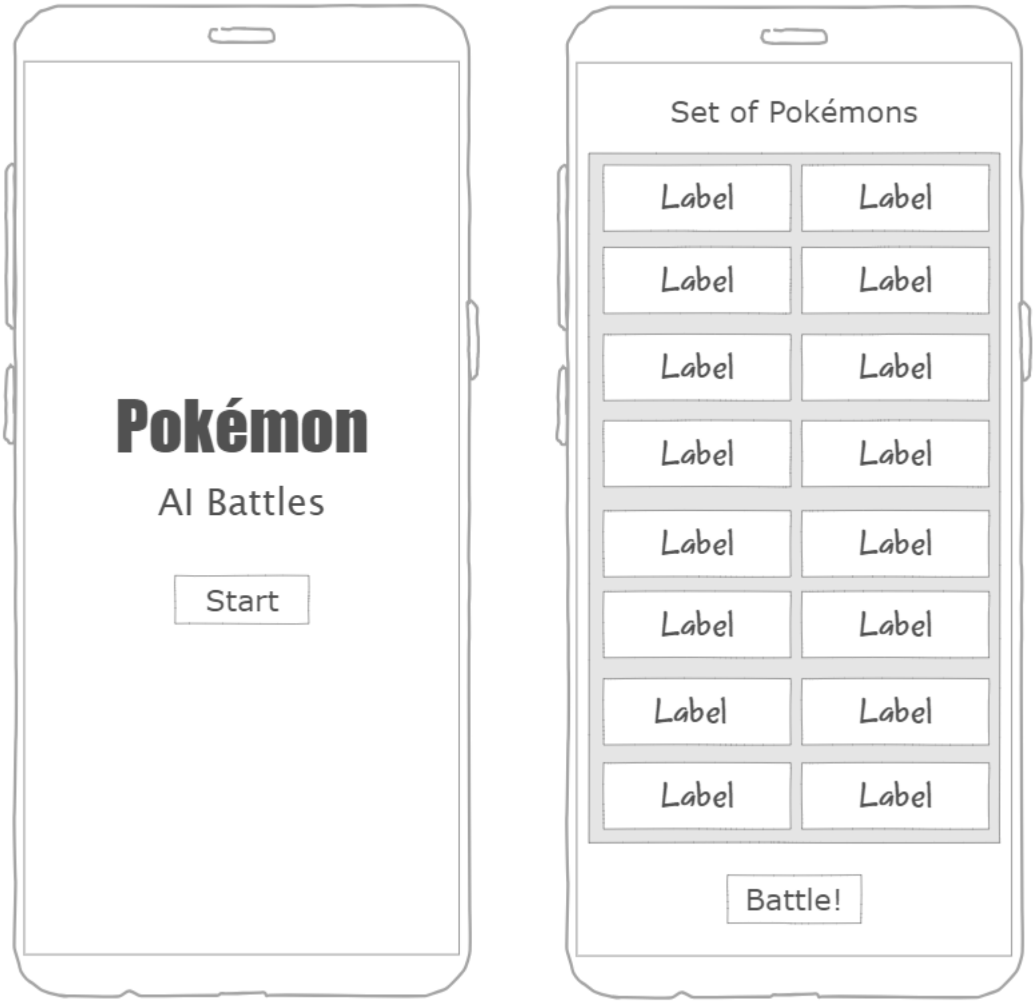
- Android Developers http://developer.android.com/index.html
- Android Developer NanoDegree https://www.udacity.com/course/android-developer-nanodegree--nd801
- Programming Mobile Applications for Android Handheld Systems: Part 1 https://www.coursera.org/learn/android-programming
- Programming Mobile Applications for Android Handheld Systems: Part 2 https://www.coursera.org/learn/android-programming-2
- Android programming course: learn how to create your own applications http://www.sgoliver.net/blog/curso-de-programacion-android/

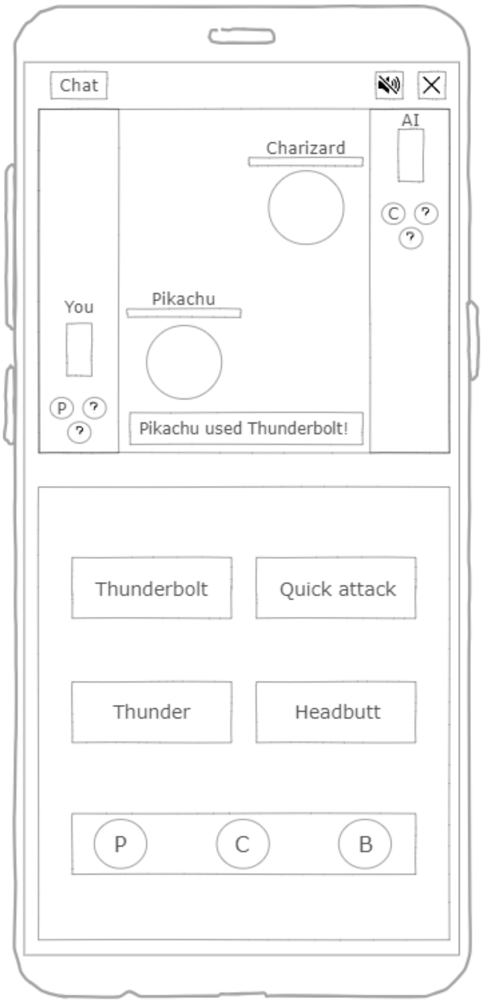
Class concepts

Application of concepts covered in class in the development of the project.

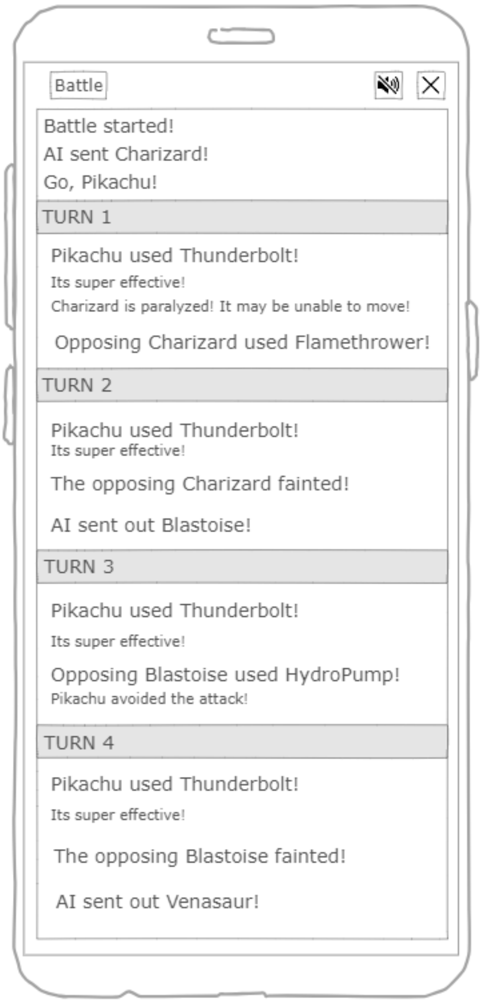
Mocks

Initial mocks or sketches hand made or done using webs like https://ninjamock.com/





Pikachu used Thunderbolt!



Battle started!
AI sent Charizard!
Go, Pikachu!

TURN 1

Pikachu used Thunderbolt!
Its super effective!
Charizard is paralyzed! It may be unable to move!
Opposing Charizard used Flamethrower!

TURN 2

Pikachu used Thunderbolt!
Its super effective!
The opposing Charizard fainted!
AI sent out Blastoise!

TURN 3

Pikachu used Thunderbolt!
Its super effective!
Opposing Blastoise used HydroPump!
Pikachu avoided the attack!

TURN 4

Pikachu used Thunderbolt!
Its super effective!
The opposing Blastoise fainted!
AI sent out Venasaur!

Individual work

From each of the members of the group, a list with: date, time taken and work done.
The information in this page will not be used to grade the students, just to estimate the amount of work done for this course.

Nicolás:

- Creation and development of the MainActivity, which is the first view of the application.(2 hour)
- Creation and development of the OptionsActivity, which is the view that allows choosing options such as viewing the Pokémon set, going to the combat view, and selecting Pokémon for battle.(3 hours)
- Creation and development of the AllPokemonActivity, which allows viewing available Pokémon and their respective statistics and types by long-pressing on them.(4 hours)
- Creation and development of the ChooseTeamActivity, which enables the selection of 3 Pokémon for battle and a button to start the battle.(5 hours)
- In the BattleActivity, integration of the core in Android Studio.(18 hours)
- Development in the BattleActivity of all the information about the buttons and Pokémon displayed when holding down on them.(6 hours)

Mateo:

- Creation and development of the FinalActivity.(2 hours)
- Creation and development of the BattleActivity (where all the game logic takes place). (20hours)
- Development of all animations in the BattleActivity.(8 hours)
- In the BattleActivity, integration of the core in Android Studio.(10 hours)

Juan:

- Development of the core of the application (game mechanics and structure without AI) (14 hours).
- Research and basic development of MCTS (Monte Carlo Tree Search) (8 hours).
- Adaptation of the basic MCTS to our game (6 hours).
- Testing and correction of errors (4 hours).

Javier:

- Development of the core of the application (game mechanics and structure without AI) (10 hours).
- Research of related literature and basic development of MCTS (Monte Carlo Tree Search) (6hours).

- Adaptation of the basic MCTS to our game (6 hours).
- Search of new metrics for the battle state (2hours).
- Testing and correction of errors (12 hours).

References

List of references used to develop the project.