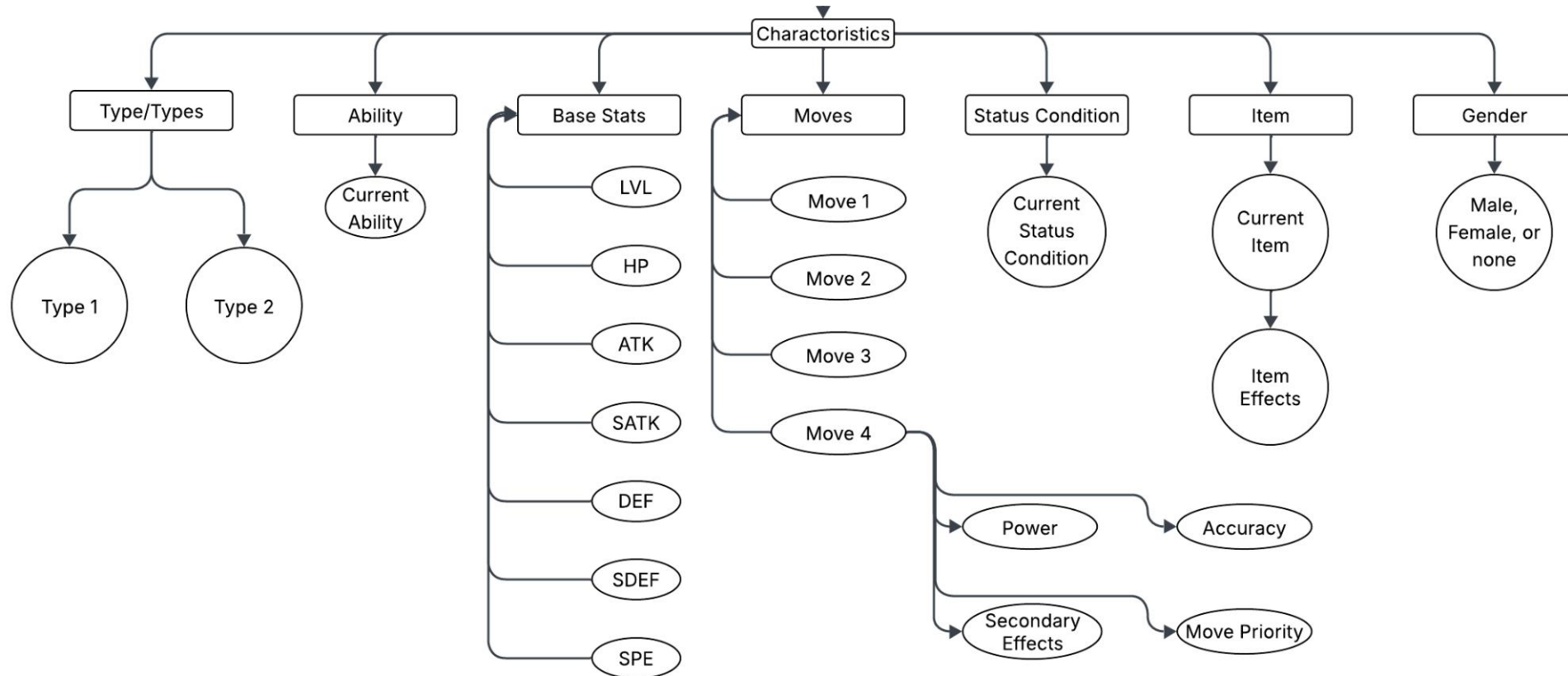
A series of white, overlapping geometric lines and polygons on a black background, located on the left side of the slide. The lines form various shapes, including triangles and quadrilaterals, some of which are nested or intersecting.

ADVANCED POKÉMON BATTLE SIMULATOR

By: William Pitts and Caleb Oglesby

WHAT MAKES A POKÉMON A POKÉMON?



IMPORTANT POKÉMON CHARACTERISTICS

1. Types
 1. Type 1: Fire
 2. Type 2: Null
2. Ability
 1. Can increase stat's
 2. Alter moves
 3. Alter Pokémon type
 4. Etc.
3. Stats
4. Moves
 1. 4 per Pokémon
 2. Setup, recovery, and attacking moves
5. Status Condition
 1. Is it burned, poised, asleep, frozen, paralyzed
 1. Some of these have effects besides their primary effect.
6. Item
 1. Works like a second ability if held like Pokémon
7. Gender
 1. Only matter is the move "Attract" is used



Item: Silk scarf increases
normal type move attack
power

TYPE AND DAMAGE INFO

Pokémon can have up to two types, and their effectiveness multipliers are affected by one another. If a Pokémon is a Grass and Ice type, then a Fire attacking move is 4 times effective against it.

Other Factors to Damage:

- Item Effects
- Ability Effects
- Etc.



Pokémon Type Chart

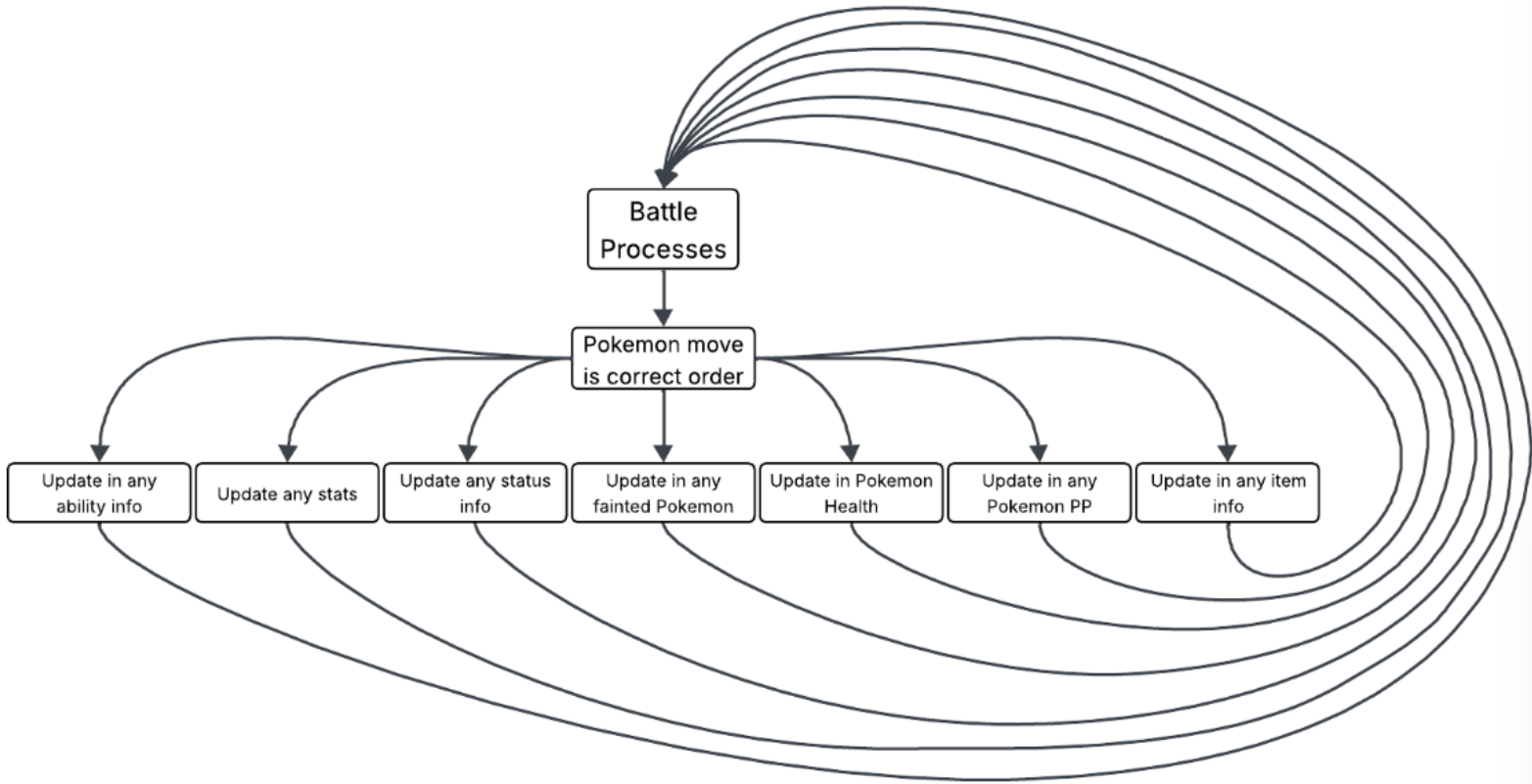
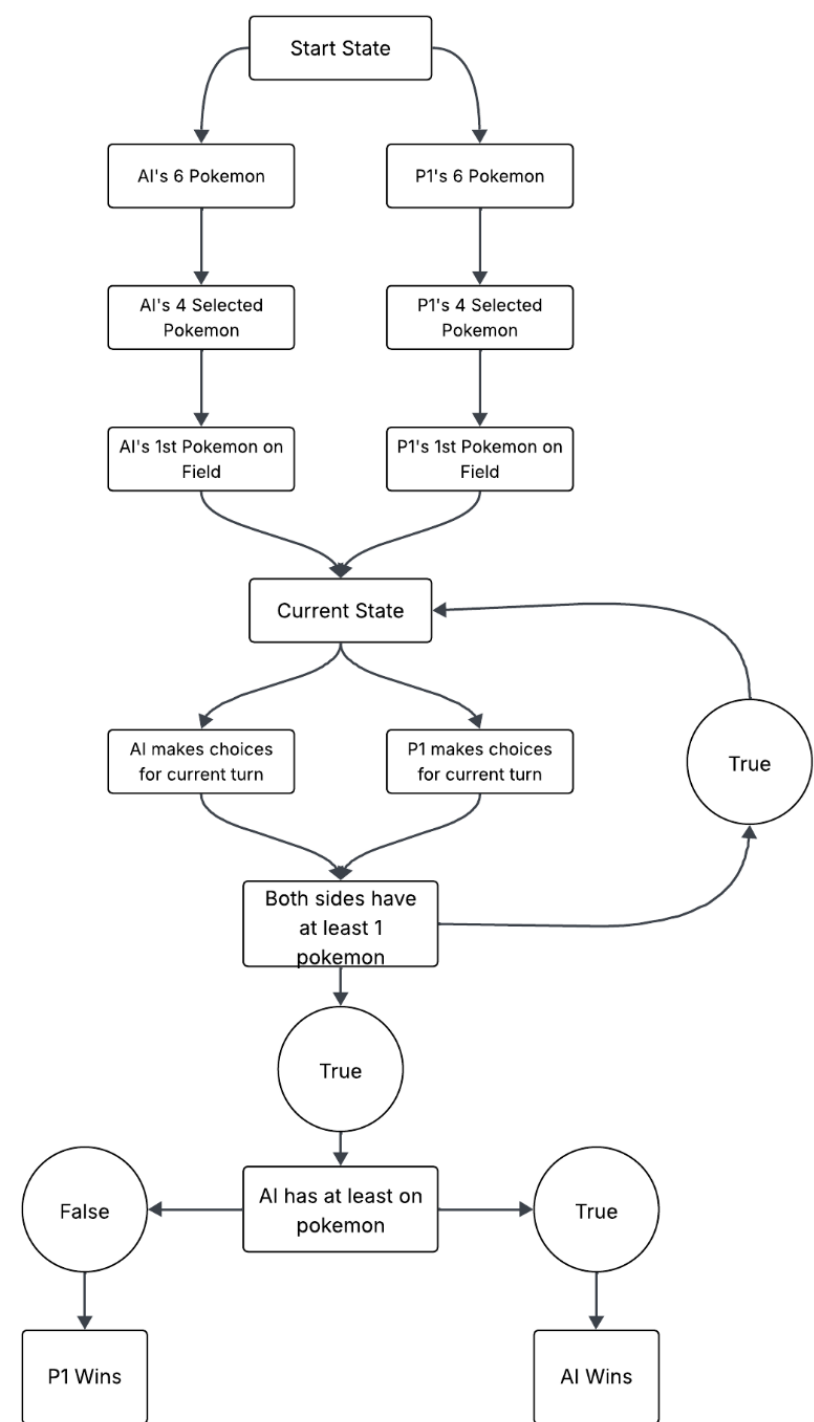
created by pokedex.net

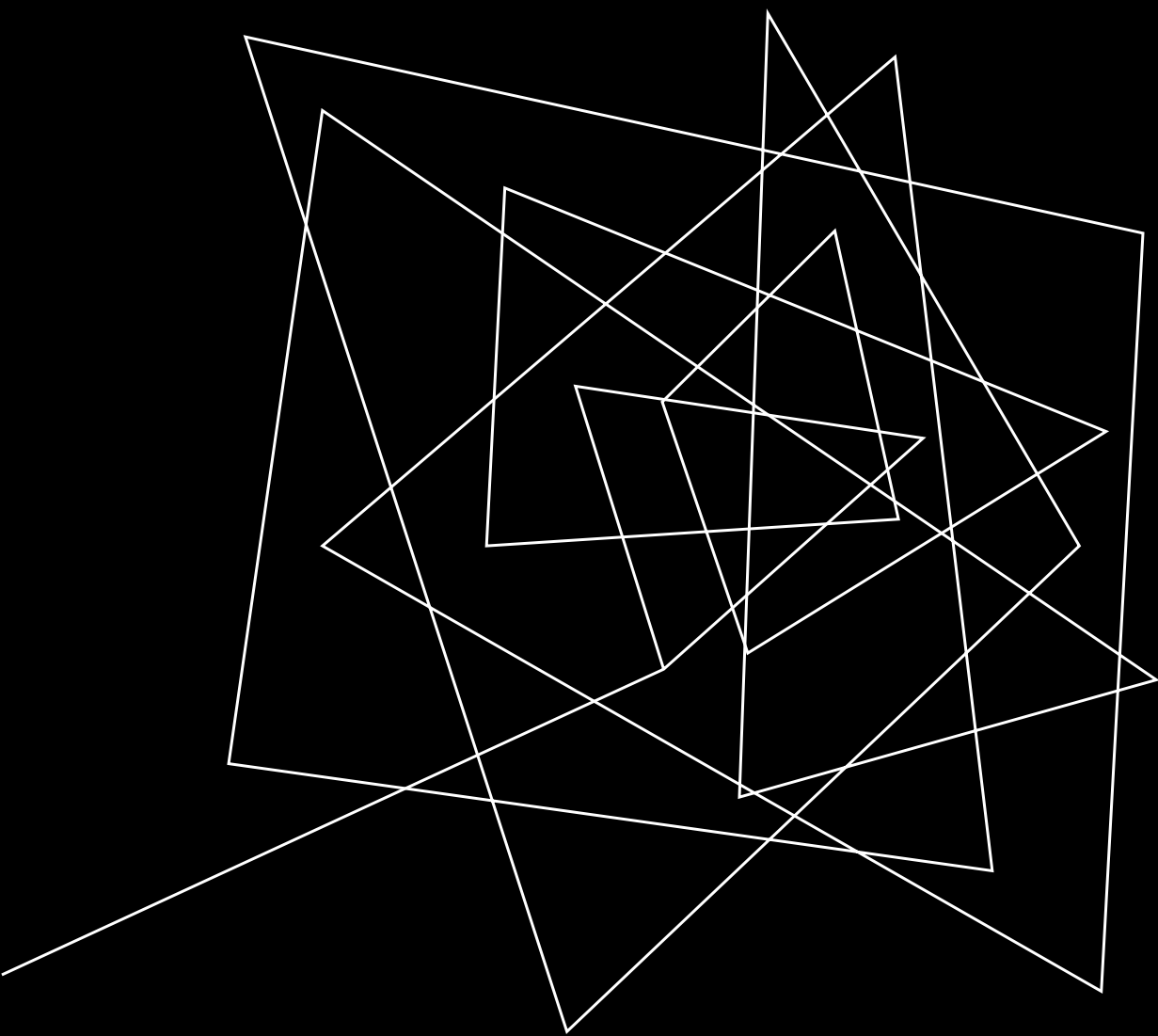
Applies to all games since Pokémon X&Y (2013)

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	DAR	STE	FAI
NORMAL													½	0			½	
FIRE		½	½		2	2						2	½		½		2	
WATER		2	½		½				2				2		½			
ELECTRIC			2	½	½				0	2					½			
GRASS		½	2		½			½	2	½		½	2		½		½	
ICE		½	½		2	½			2	2					2		½	
FIGHTING	2					2		½		½	½	½	2	0		2	2	½
POISON					2			½	½				½	½			0	2
GROUND		2		2	½			2		0		½	2				2	
FLYING				½	2		2					2	½				½	
PSYCHIC							2	2			½					0	½	
BUG		½			2		½	½		½	2			½		2	½	½
ROCK		2				2	½		½	2		2					½	
GHOST	0										2			2		½		
DRAGON															2		½	0
DARK							½				2			2		½		½
STEEL		½	½	½		2							2				½	2
FAIRY		½					2	½							2	2	½	

$$Damage = \left(\frac{\left(\frac{2 \times Level}{5} + 2 \right) \times Power \times \frac{A}{D}}{50} + 2 \right) \times Targets \times PB \times Weather \times GlaiveRush \times Critical \times random \times STAB \times Type \times Burn \times other \times ZMove \times TeraShield$$

HOW DOES A SINGLES POKEMON BATTLE WORK





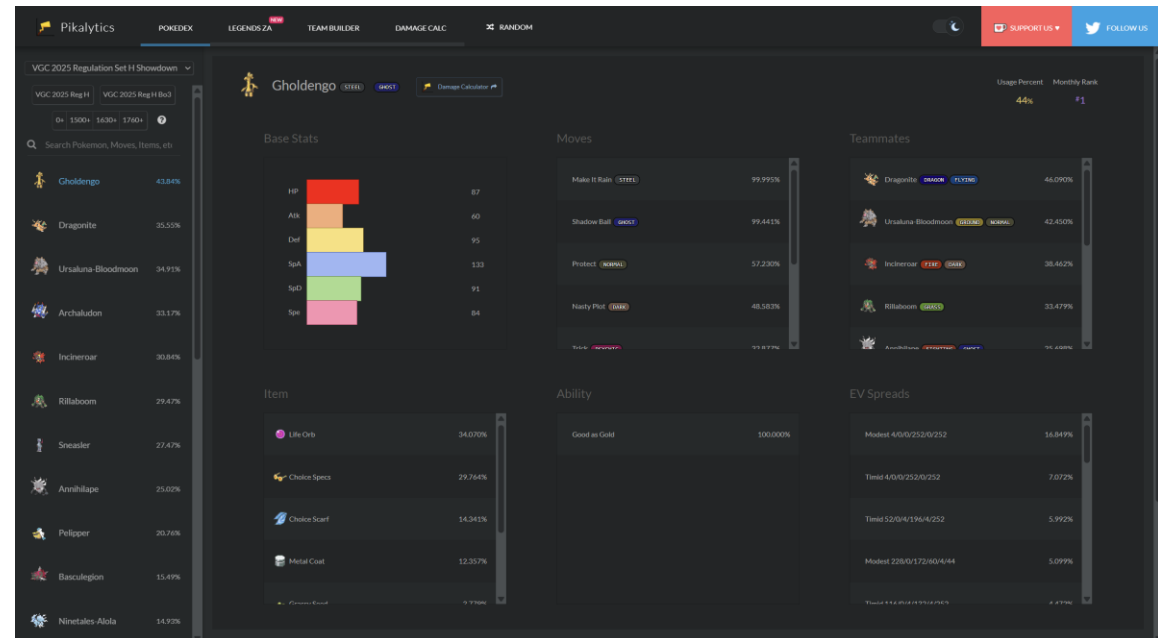
DATA LOADING AND PROCESSING

How we used this website

1. We scanned all HTML for every Pokémon page by using two files (generate_pikalytics_compendium.py and pikalytics.util.py):
 - a. Adding /Pokémon_Name to “<https://www.pikalytics.com/pokedex/gen9vgc2025regh>”
2. It would then generate separate HTML files for each Pokémon
3. These files would later be used to as the source of important data like what percentage of time a Pokémon is used Competitively, what moves it uses, etc.
4. That is done by our Data loader script.

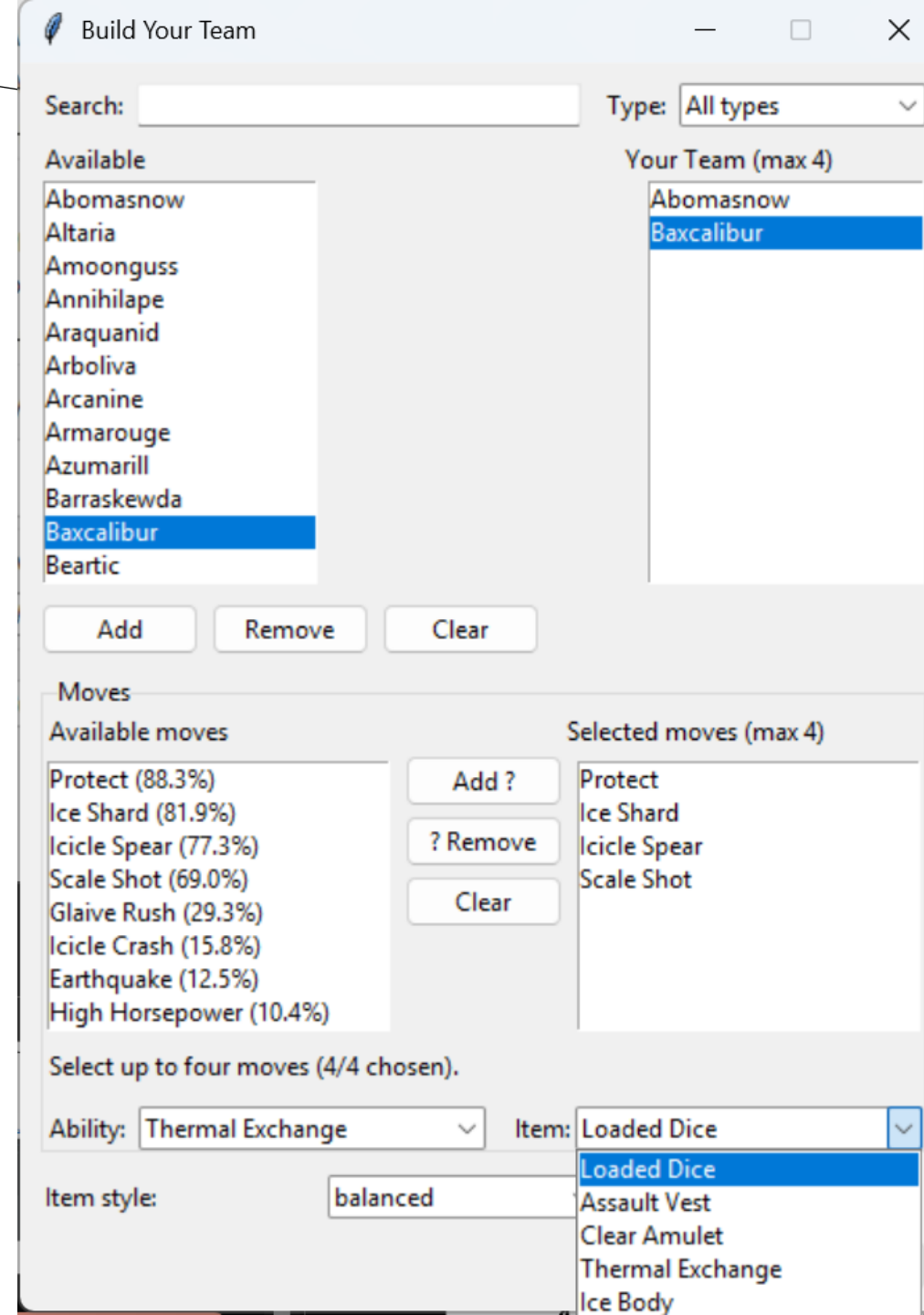
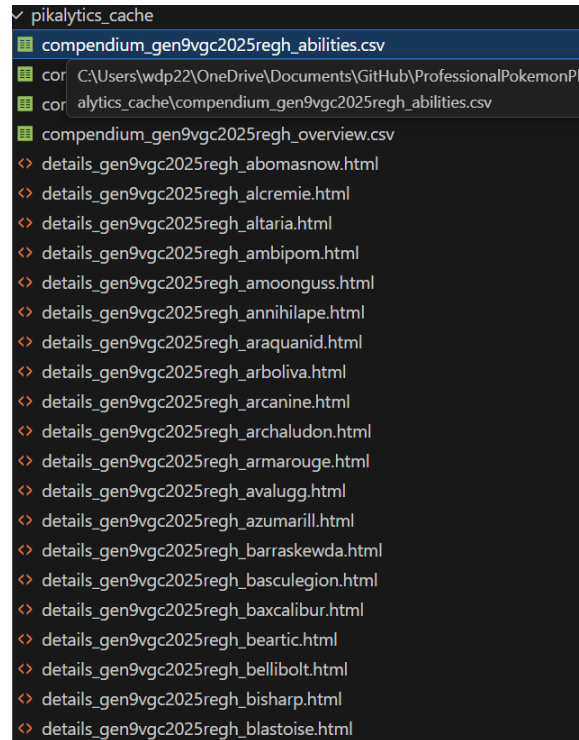
PIKALYTIC.COM

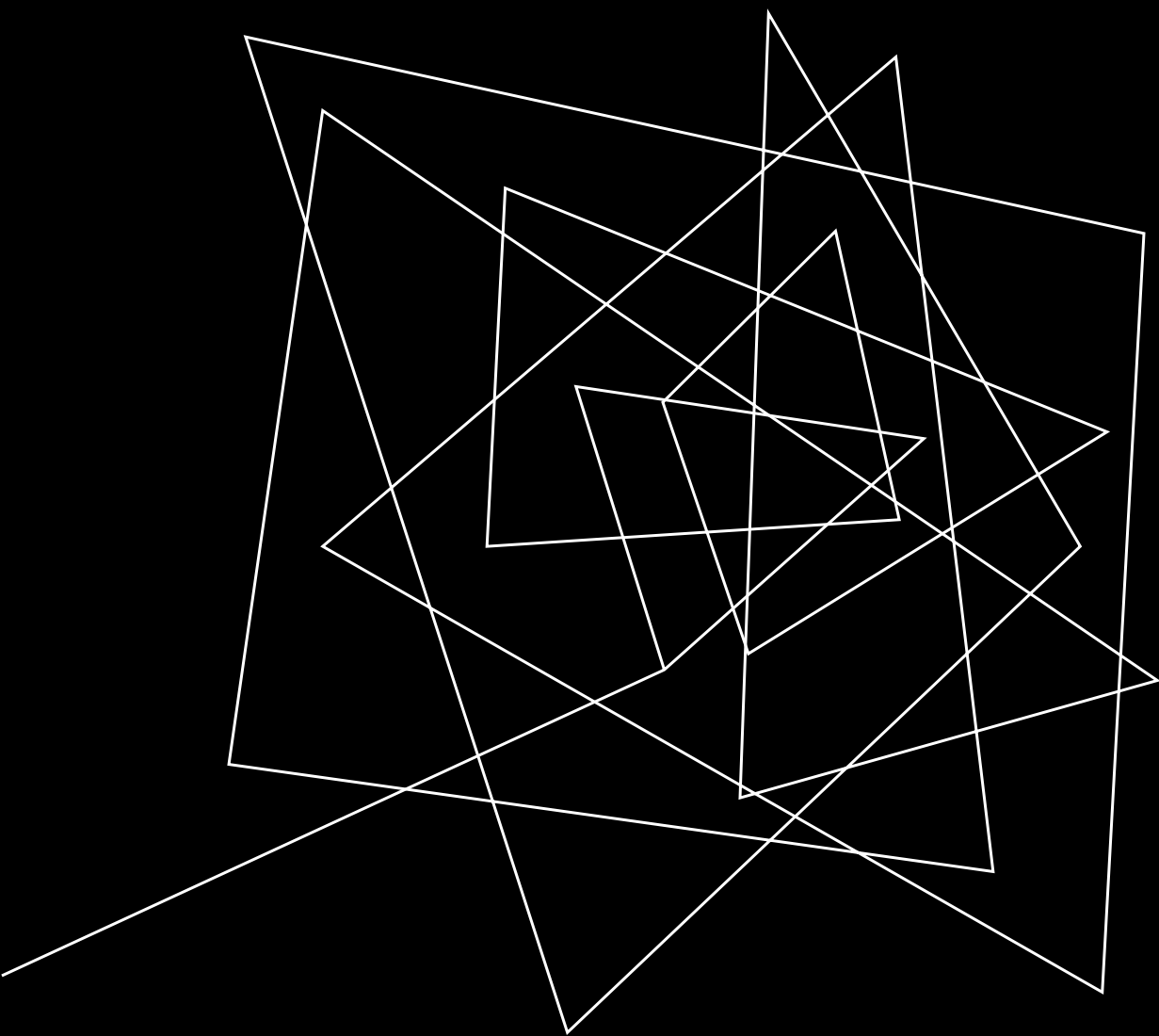
<https://www.pikalytics.com/pokedex/gen9vgc2025regh>



DATA LOADER

1. Data Loader read several important csv files using pandas so it can better parse through the HTML files.
2. It has key words that it will look for within the HTML
3. Data Loader enables Team Builder AI Battler to function using the algorithm we designed, while also making Team Builder GUI for the Player More understandable and helpful for a competitive team builder.
4. Pikalytics files and Data Loader scripts are the most important and unique files we made.

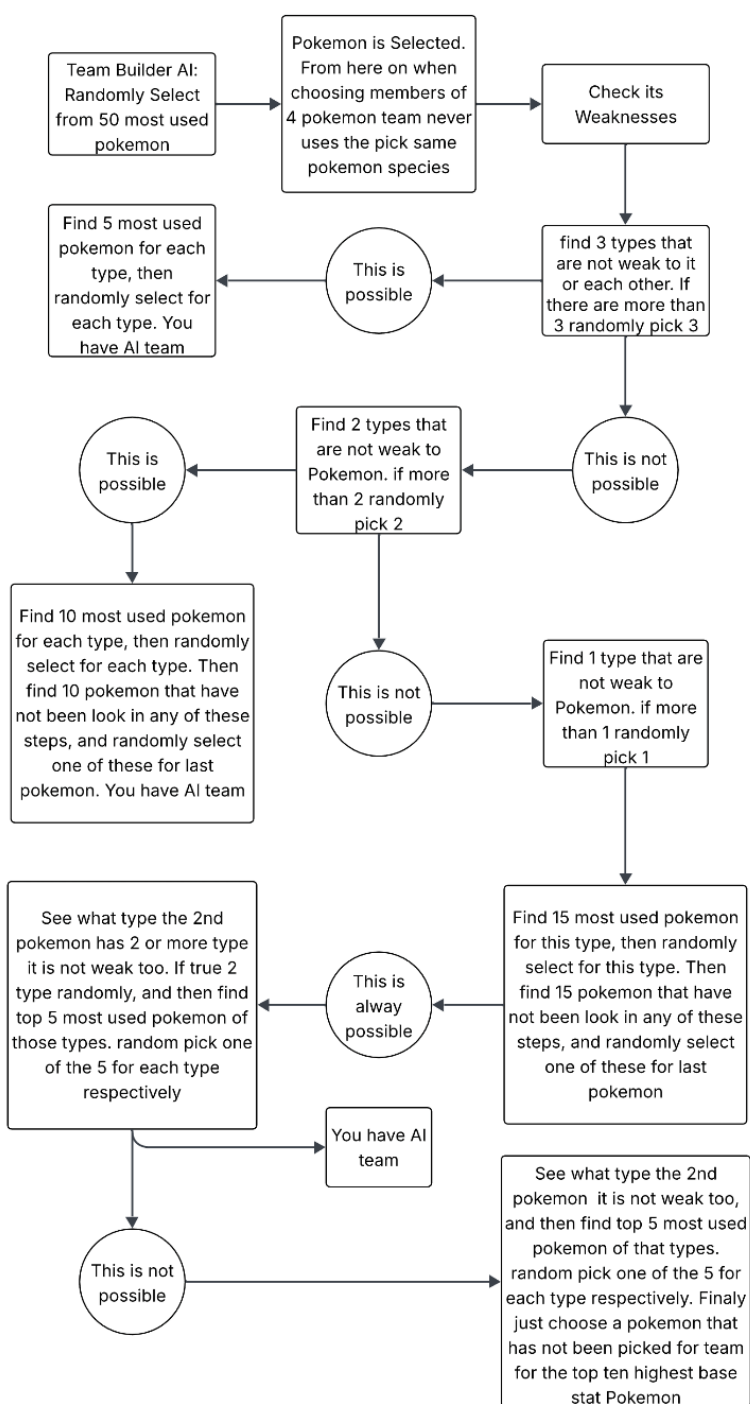




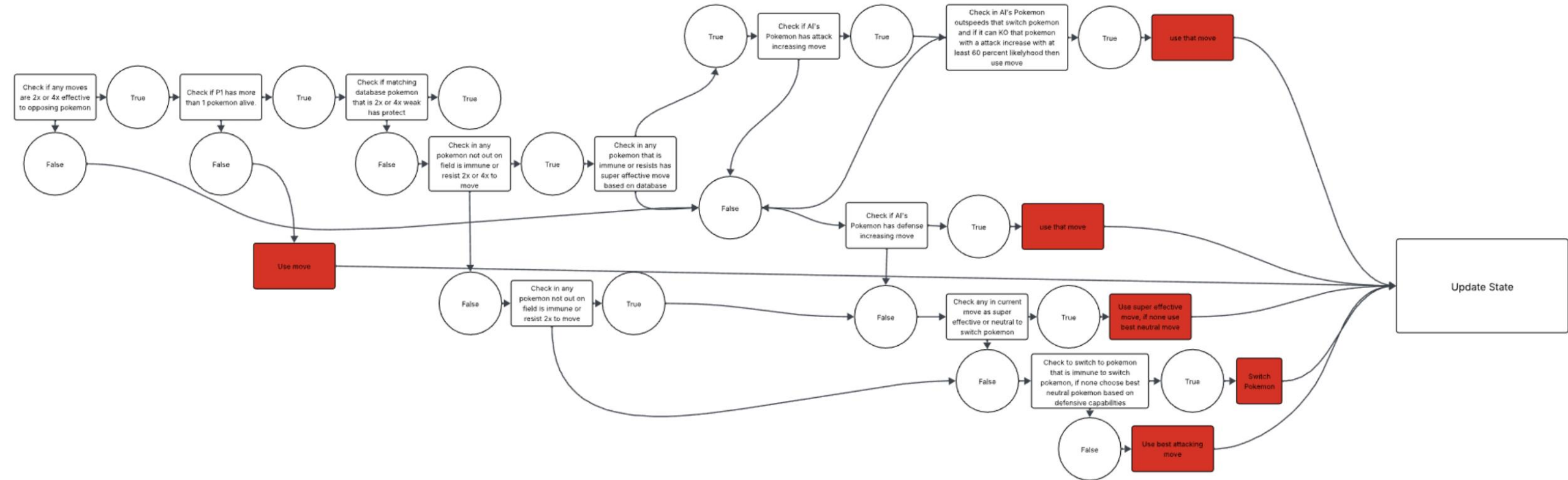
AI TEAM BUILDER AND BATTLER ALGORITHMS

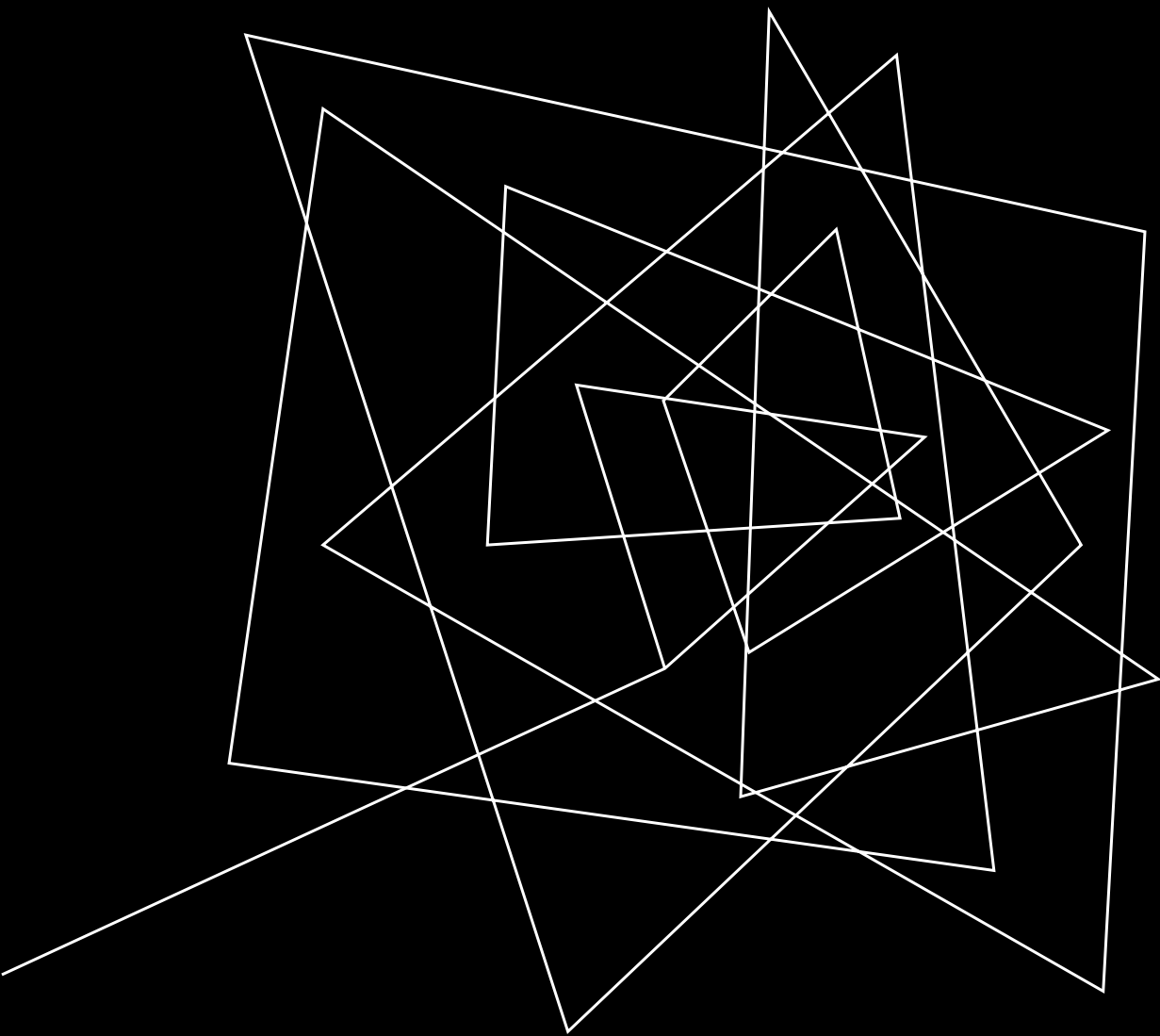
TEAM BUILDER

When we were designing the team builder, our original Algorithm would always poke the same Pokémon. Since it was deterministic, we used the statistics from Pikalytics to design a team builder for the AI to used with elements of randomized selection. We mad it pick the most used Pokémon because they are typically the best Pokémon while making sure to pick complementary types.



AI BATTLER ALGORITHM





Now Let's Run
our Application!