Detailed Design of Solution

Input

Input	Data Type	Normal Range etc.	Example
Pokemon Name	String	N/A	Pikachu
Game Text	String	N/A	Battle started between Orareo and zgjoj! Go! **Hydreigon**! zgjoj sent out **Hippowdon**! [The opposing Hippowdon's Sand Stream] A sandstorm kicked up!
Advice Note	String	N/A	“Keep Necrozma Dawn-Mane Alive and switch it into opponent’s Xerneas”
Name of Note	String	N/A	“Frosmoth Sweep 1”
Important Pokemon	String	N/A	“Ferrothorn, Dragapult”
Threats	String	N/A	“Excadrill, Cinderace”
Opponent predict Note	String	N/A	“They will use a substitute, then set up”
Moves	String	N/A	“Thunder”
Item	String	N/A	“Life Orb”
Ability	String	N/A	“Lightning Rod”
Your Pokemon	String	N/A	“Hydreigon”

Output

Advice	String	N/A	“Hit the opponent with a special attack move”
Predict	String	N/A	“The opponent will probably attack it”
Important Pokemon	String	N/A	“Ferrothorn, Dragapult”
Threats	String	N/A	“Excadrill, Cinderace”
Opponent predict Note	String	N/A	“They will use a substitute, then set up”
Advice Note	String	N/A	“Hit it hard with a special attack move”
Opponent’s Pokemon Base Stats	Integer	1 - 4	100
Pokemon Weakness	String	N/A	“Ghost, Dark”
Advice	String	N/A	“Hit the opponent with a special attack move”

Original Prototype

Move Adviser | Damage Calculator | Notes

Input Texts

Paste the game text here.

Your Team

Import your team here.

Predicted Opponent's Next Move

The move your opponent will probably do next

Your Advised Next Move and Reminder

The move you should do next and notes about it

Important Pokemon

Pokemon you should keep alive and healthy

Move Adviser | Damage Calculator | Notes

Damage on Opponent

x %

Your Type

Normal | ----

Your Move Type

Normal | ATK

Move Damage

Move Damage Multiplier

1

Import Your Pokemon

Offensive Stat Multiplier

1

Defensive Stat Multiplier

1

Reflect/Light Screen

Aurora Veil

Dynamax | Crit

Weather

none

Terrain

none

Type Chart

	N	FR	W	E	GS	I	FT	P	GD	FL	PS	B	R	GH	DR	DK	S	FA
N													1/2	0			1/2	
FR		1/2	1/2		2	2						2	1/2		1/2		2	
W		2	1/2		1/2				2				2		1/2			
E			2	1/2	1/2				0	2					1/2			
GS		1/2	2		1/2		1/2	2	1/2		1/2	2			1/2		1/2	
I		1/2	1/2		2	1/2			2						2		1/2	
FT	2					2		1/2		1/2	1/2	1/2	2	0		2	2	1/2
P					2			1/2	1/2				1/2	1/2			0	2
GD	2		2	1/2			2		0			1/2	2				2	
FL				1/2	2		2					2	1/2				1/2	
PS							2	2			1/2					0	1/2	
B		1/2		2		1/2	1/2		1/2	2				1/2	2	1/2	1/2	
R		2				2	1/2		1/2	2		2					1/2	
GH	0										2			2		1/2		
DR															2		1/2	0
DK							1/2				2			2	1/2		1/2	
S		1/2	1/2	1/2		2							2				1/2	2
FA		1/2					2	1/2							2	2	1/2	

Damage on You

x %

Opponent's Type

Normal | ----

Opponent's Move Type

Normal | ATK

Move Damage

Move Damage Multiplier

1

Import Opponent's Pokemon

Offensive Stat Multiplier

1

Defensive Stat Multiplier

1

Reflect/Light Screen

Aurora Veil

Dynamax | Crit

Move Adviser | Damage Calculator | Notes

Your Pokemon

Import

Record Moveset

What You Should Do

Possible Swaps

Important Counter Pokemon

Explain

Opponent's Pokemon

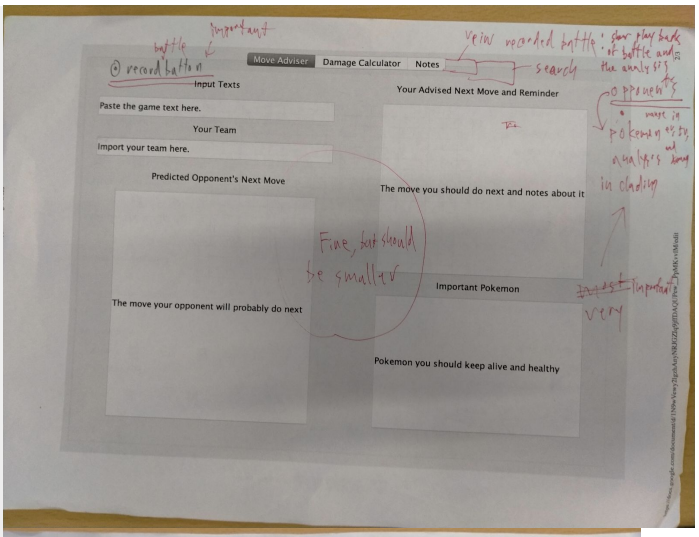
Import

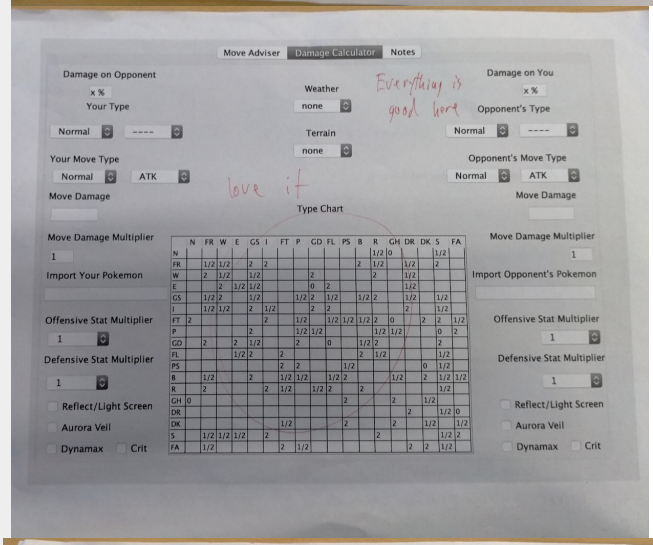
Record Moveset

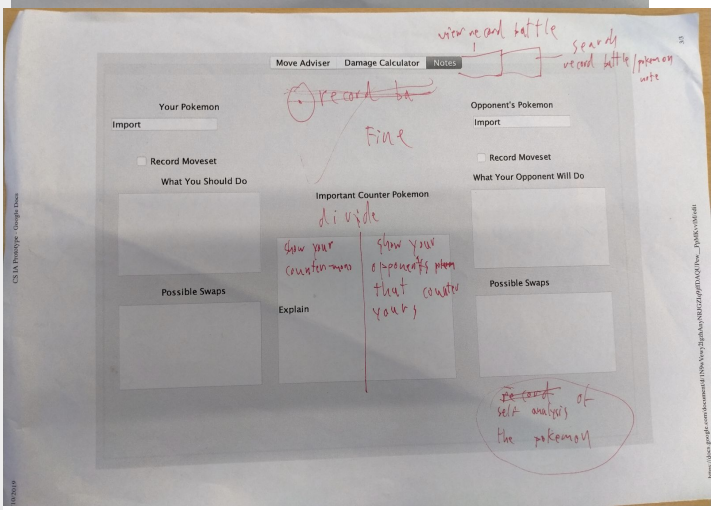
What Your Opponent Will Do

Possible Swaps

Prototype Interview Picture





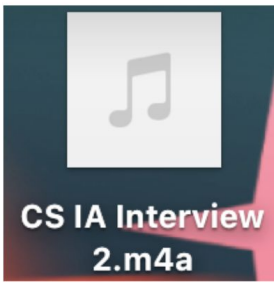


Interview Audio and Transcript

Me: Ok Julian what's the thing you that you think should change or like, do you want me to add anything?
Julian: For your team here, is this just for the program? Or are you ganna have like a drop-down where it shows like Pokemon on your team.
Me: Ah, no, there's an import feature for Pokemon Showdown, like paste it down here and it will like, do its stuff.
Julian: So, are you gonna add lie, so you can see-
Me: It's just like an input, yup. You take in the calculation stuff from your team and stuff like that.
Julian: Does this also take into account like opponent's Pokemon like, stats and move and items as well?
Me: The move and item you have to put it in by yourself to the damage calculator, maybe hold, the notes here...maybe I should add a feature where, actually, putting every item into the database would be a heck ton of, like, thing so I think this part you have to like, note it in by yourself what you do but, the move advisor plan on like making a basic common sense program where you predict the basic move and stuff like that. But I plan on putting every Pokemon's stats and stuff into the database.
Julian: So the note is like, you can type yourself? So the program doesn't put anything for you like what you should do is it is this like a move recommender or you have like, the move advisor will you also have the "what you should do", like what's the difference between and the move advisor?
Me: Oh the move advisor, basically the thing that you put here and the thing that will come out the calculation based on the team that you put here and database based on your opponent's Pokemon. The thing here, if you click record moveset of that Pokemon it'll record the moveset and stat of that Pokemon and improve on the move advisor. I'll try to make it like that so it will use the feature of the damage calculator and see whether what you should do.
Julian: You know what I didn't like is the record moveset thing, like, are you able to go back and see the move you make and the move your opponent makes to see the possible mistake you did and the pattern of your opponent, cuz if there's a playback feature where you can go over and see.
Me: That's a good idea
Julian: Or the record moveset just for in-game, so it can predict on what you should do.
Me: At first this feature is for the record just to predict the move it does but I think you have a point I should make it like that so you can record the battle and see which move you can do better and stuff like that
Julian: umm, and so possible swap is this like um, manual notes or is this from a recommender?
Me: It'll be from the recommender, when a battle starts it will give the predict of the team of the opponent also right so if I put the database of every Pokemon in here it will take in account of all the type chart and predict the basic ones
Julian: So how did you like, make it so cuz you can customize the stats of the Pokemon that you used to make like whether you wanted it to be special attacker or physical attacker like how did you incorporate that when you insert the team.
Me: I'm planning on looking up online on how to separate the text of the team but on the thing about whether it is a physical damage or special damage that part should be easy because if I can separate the text, like what it should take in then, I can set them as different value and use this value for attack and this value for calculating a special attack and something like that.
Julian: will it tell you on the app whether the opponent Pokemon that you're going against, like, their strength and their weaknesses, like do they have a good special defense or physical defense, whether they're a glass cannon
Me: Make sense, I should add that as like, another advisor window about an analysis of that Pokemon
Julian: Yeah, if that's possible
Me: yeah if it's possible, maybe I should add like, Smogon's analysis too. Maybe that might not be possible but I'll try. But at least that should be the analysis of the stat.
Julian: Yeah. So so for the important counter Pokemon, does it take Pokemon from your team?
Me: Yes
Julian: And like, does it show which Pokemon on their team counter your Pokemon? Or no
Me: ah, no, it's just for note, like, which Pokemon you need to keep alive to counter that Pokemon, such as keep your Rotom-wash alive and something like that.
Julian: I really like this page (damage calculator), though it was done super well.
Me: Oh yeah, should I put the type in another tab? Or is it fine in the damage calculator
Julian: I think the type chart is good in the damage calculator cuz it fits under… I mean when I play I always have another tab open for type chart. It's good that it's right here so I can see. I think this is done super well. And that was all of my initial comments. I really like the predict, I thought this was done super cool, and the advisor thing, my main thing that I think would be cool is if there is a playback feature
Me: Ok, I'll note that down
Julian: Like, if you can add a playback feature and like, a basic analysis of the Pokemon
Me: Ok, playback feature and the analysis of the Pokemon. Should I add like, analysis of your Pokemon also? Or show the base stats of your Pokemon and the opponent's Pokemon.
Julian: Should it? Like, you could, I think the analysis of the Pokemon is more like, for beginners, you know like me who doesn't necessarily always know the stats of the Pokemon but you should know the stats of your own.
Me: Ok, so only in-depth analysis of the opponent's Pokemon like the range in EV and IV and the range in damage and stuff like that
Julian: Yeah, I think that would be… if that's possible it would be really cool, and the record, like the playback
Me: Ok, I should be able to do that. Alright, thanks.

Audio link of the initial interview: https://chirb.it/44DzOK

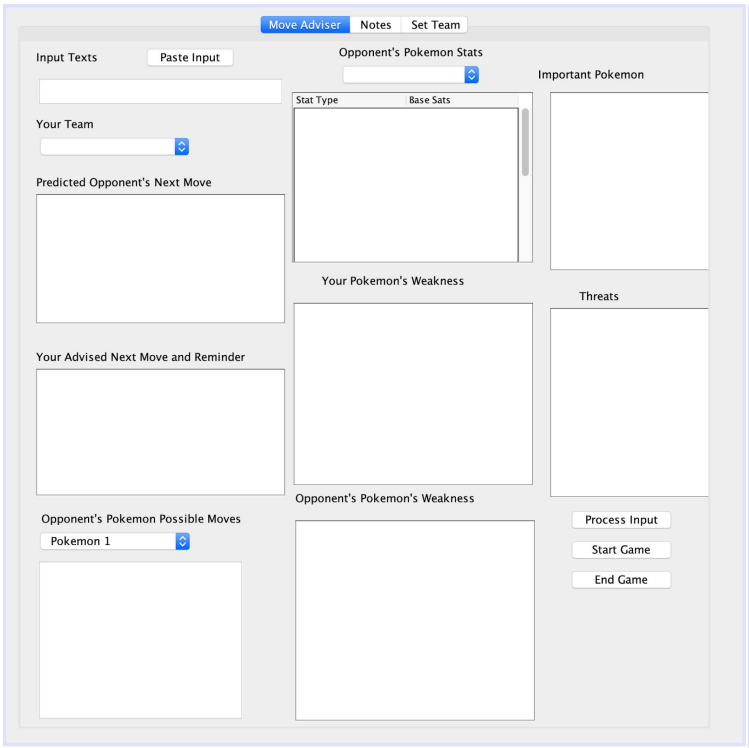
Audio link of the prototype interview: https://chirb.it/PK6Bp5

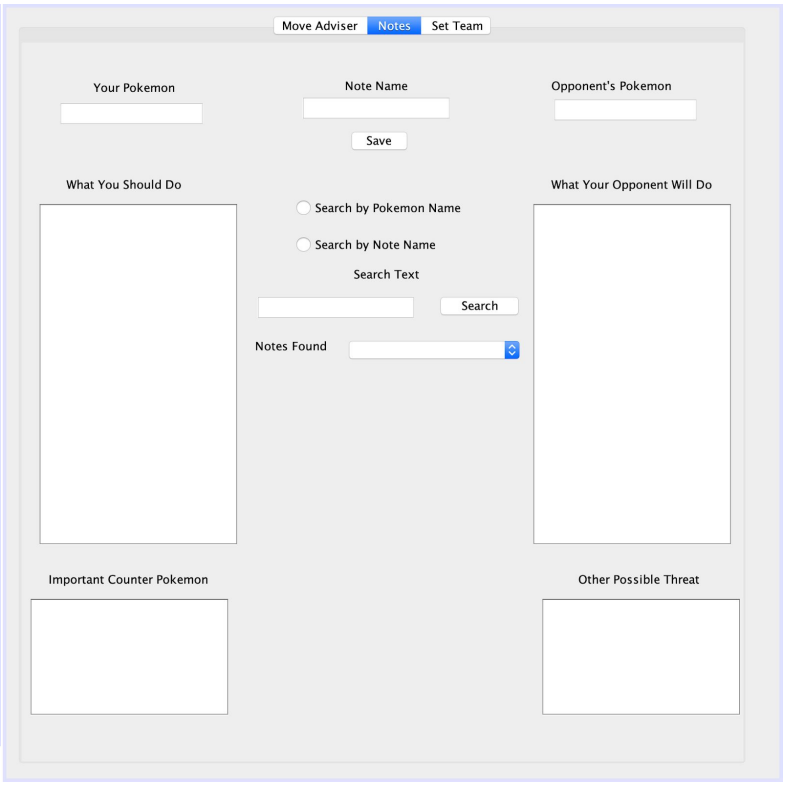


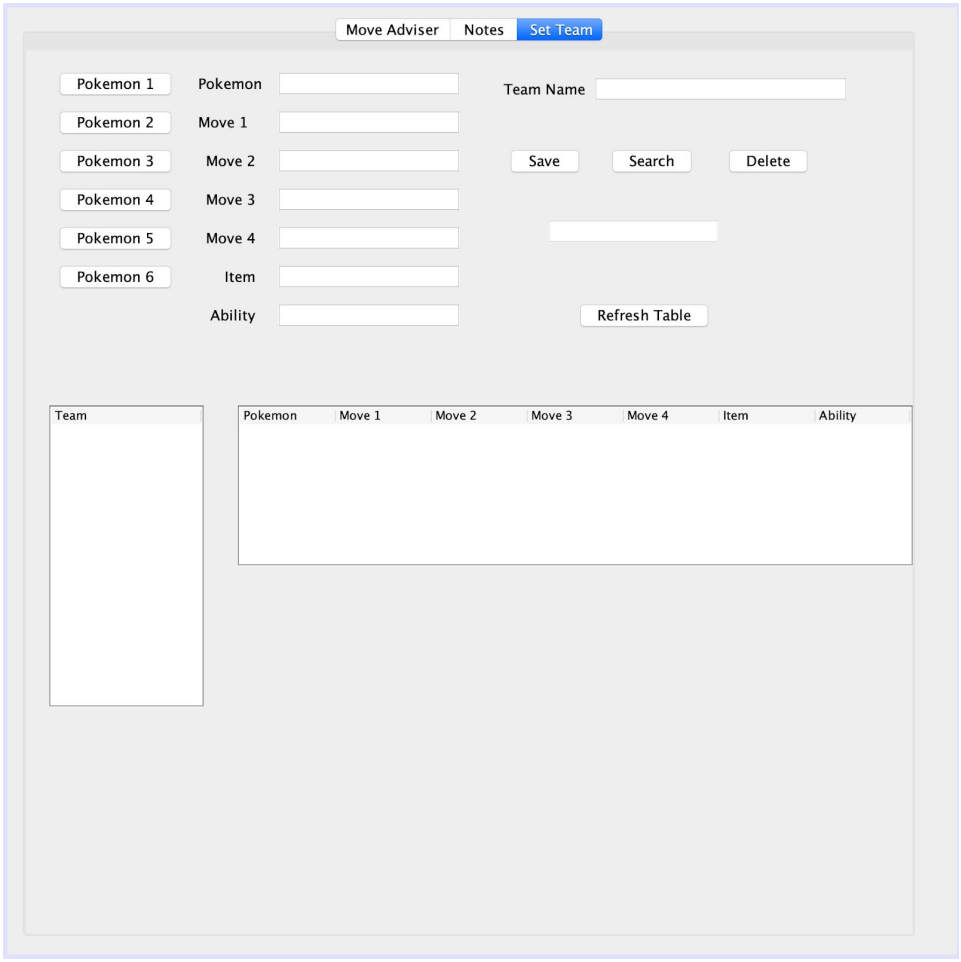
Proposed Changes

- Have a feature to record and view the battle replay
- Have a feature to analyze the opponent's Pokemon

Final Prototype







First Panel: Move Advisor

- Deleted damage calculator panel
- Replaced type chart with weakness text fields

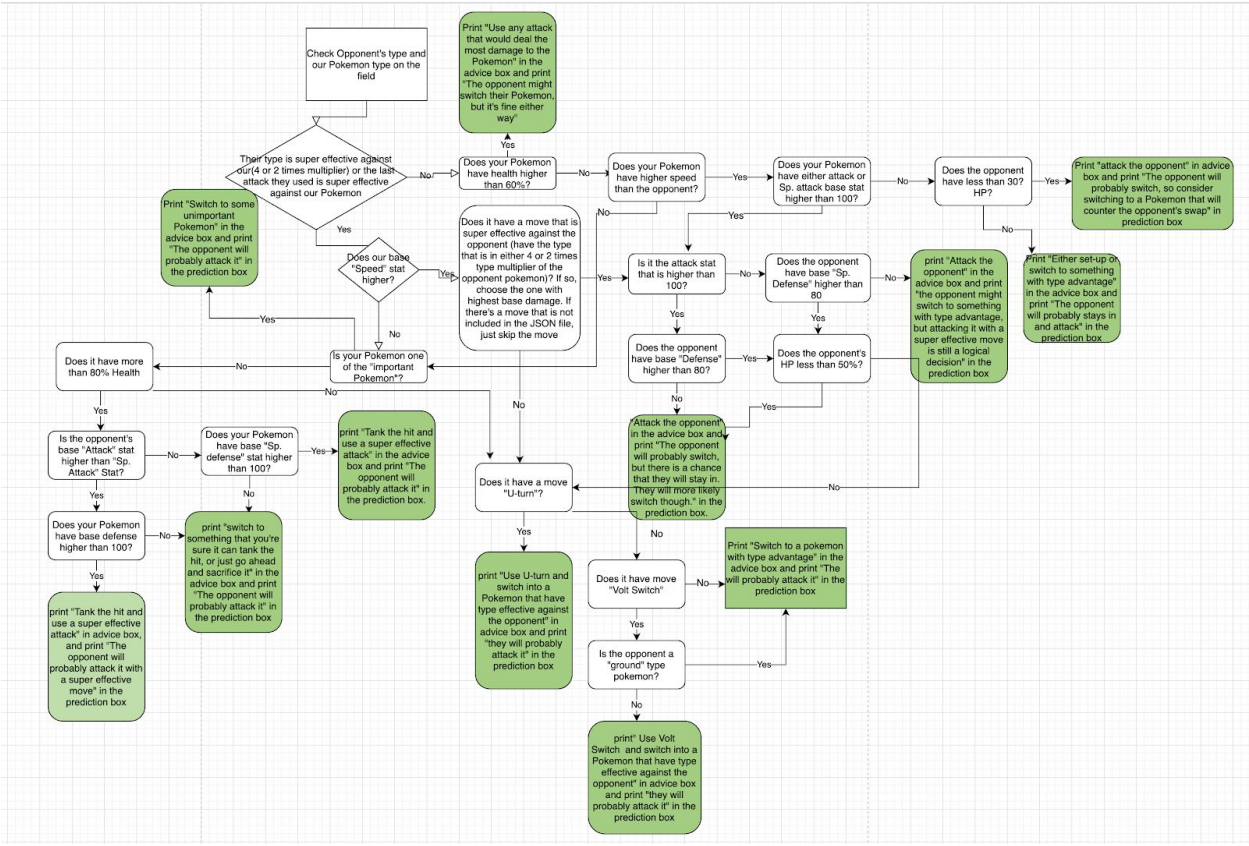
Second Panel: Notes

- Included threat Pokemons
- Can view/save notes

Third Pane: Set Teams

- Added team saving and searching feature

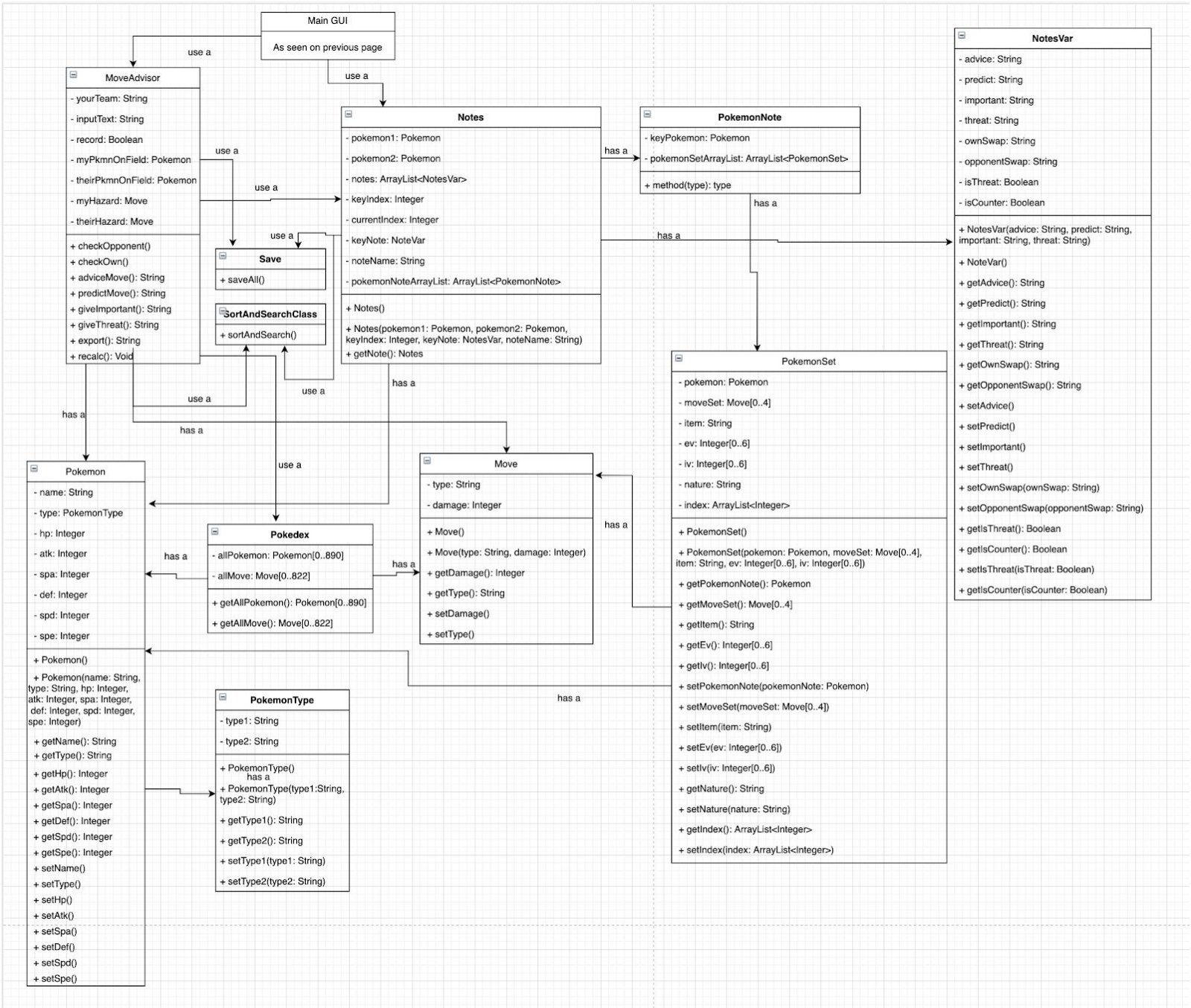
Flowchart of Move Advisor Algorithm



UML Diagram of the MainGUI

The program have a lot of stuff in its GUI.

UML Diagram


Main GUI
As seen on previous page
use a
MoveAdvisor
- yourTeam: String
- inputText: String
- record: Boolean
- myPkmnOnField: Pokemon
- theirPkmnOnField: Pokemon
- myHazard: Move
- theirHazard: Move
+ checkOpponent()
+ checkOwn()
+ adviceMove(): String
+ predictMove(): String
+ giveImportant(): String
+ giveThreat(): String
+ export(): String
+ recalc(): Void
Notes
- pokemon1: Pokemon
- pokemon2: Pokemon
- notes: ArrayList<NotesVar>
- keyIndex: Integer
- currentIndex: Integer
- keyNote: NoteVar
- noteName: String
- pokemonNoteArrayList: ArrayList<PokemonNote>
+ Notes()
+ Notes(pokemon1: Pokemon, pokemon2: Pokemon, keyIndex: Integer, keyNote: NotesVar, noteName: String)
+ getNote(): Notes
PokemonNote
- keyPokemon: Pokemon
- pokemonSetArrayList: ArrayList<PokemonSet>
+ method(type): type
NotesVar
- advice: String
- predict: String
- important: String
- threat: String
- ownSwap: String
- opponentSwap: String
- isThreat: Boolean
- isCounter: Boolean
+ NotesVar(advice: String, predict: String, important: String, threat: String)
+ NoteVar()
+ getAdvice(): String
+ getPredict(): String
+ getImportant(): String
+ getThreat(): String
+ getOwnSwap(): String
+ getOpponentSwap(): String
+ setAdvice()
+ setPredict()
+ setImportant()
+ setThreat()
+ setOwnSwap(ownSwap: String)
+ setOpponentSwap(opponentSwap: String)
+ getIsThreat(): Boolean
+ getIsCounter(): Boolean
+ setIsThreat(isThreat: Boolean)
+ getIsCounter(isCounter: Boolean)
Save
+ saveAll()
SortAndSearchClass
+ sortAndSearch()
PokemonSet
- pokemon: Pokemon
- moveSet: Move[0..4]
- item: String
- ev: Integer[0..6]
- iv: Integer[0..6]
- nature: String
- index: ArrayList<Integer>
+ PokemonSet()
+ PokemonSet(pokemon: Pokemon, moveSet: Move[0..4], item: String, ev: Integer[0..6], iv: Integer[0..6])
+ getPokemonNote(): Pokemon
+ getMoveSet(): Move[0..4]
+ getItem(): String
+ getEv(): Integer[0..6]
+ getIv(): Integer[0..6]
+ setPokemonNote(pokemonNote: Pokemon)
+ setMoveSet(moveSet: Move[0..4])
+ setItem(item: String)
+ setEv(ev: Integer[0..6])
+ setIv(iv: Integer[0..6])
+ getNature(): String
+ setNature(nature: String)
+ getIndex(): ArrayList<Integer>
+ setIndex(index: ArrayList<Integer>)
Pokemon
- name: String
- type: PokemonType
- hp: Integer
- atk: Integer
- spa: Integer
- def: Integer
- spd: Integer
- spe: Integer
+ Pokemon()
+ Pokemon(name: String, type: String, hp: Integer, atk: Integer, spa: Integer, def: Integer, spd: Integer, spe: Integer)
+ getName(): String
+ getType(): String
+ getHp(): Integer
+ getAtk(): Integer
+ getSpa(): Integer
+ getDef(): Integer
+ getSpd(): Integer
+ getSpe(): Integer
+ setName()
+ setType()
+ setHp()
+ setAtk()
+ setSpa()
+ setDef()
+ setSpd()
+ setSpe()
Move
- type: String
- damage: Integer
+ Move()
+ Move(type: String, damage: Integer)
+ getDamage(): Integer
+ getType(): String
+ setDamage()
+ setType()
Pokedex
- allPokemon: Pokemon[0..890]
- allMove: Move[0..822]
+ getAllPokemon(): Pokemon[0..890]
+ getAllMove(): Move[0..822]
PokemonType
- type1: String
- type2: String
+ PokemonType()
+ PokemonType(type1:String, type2: String)
+ getType1(): String
+ getType2(): String
+ setType1(type1: String)
+ setType2(type2: String)
has a
use a

Development Plan

#1 - Input Pokemon and Moves into Classes (2 - 3 days)

- Create a Pokemon and Move class
- Give Pokemon class all the attributes necessary
- Give Move class attribute of type and base damage
- Insert all the 890 and more Pokemon as an array into the class
- Use a method to fills the array made in main class with the instances of Pokemon
- Do the same thing to 822 moves

#2 - Code the Algorithms (1 week)

- Code move advisor algorithm
- Create an enemy move predict algorithm

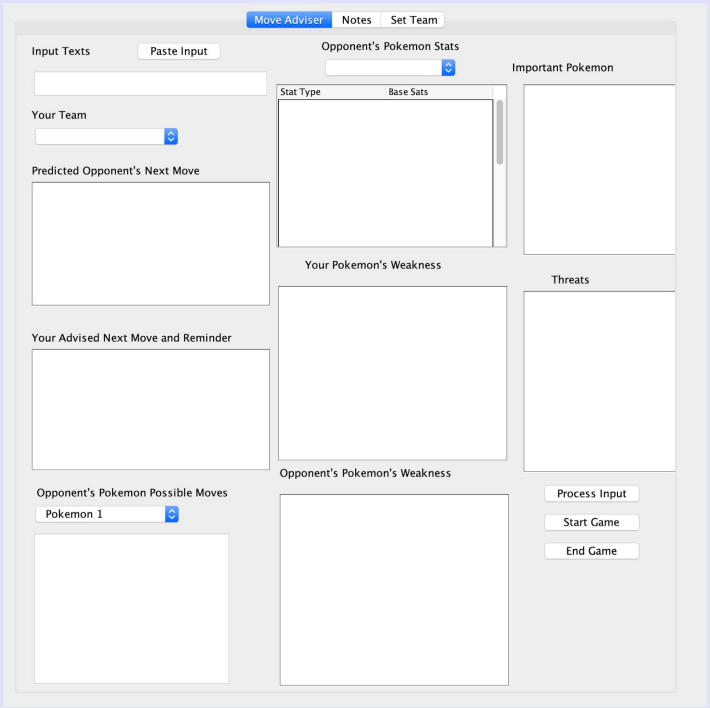
#3 - Create a Class and Methods to Store Data in Memory (1 week)

- Create a class for methods that store data from Notes and Record panel
    - Notes
        - Moveset
        - Items
        - Notes
    - Teams
        - Pokemon
        - Moveset
        - Item
        - Ability

#4 - Create Methods for GUI (2 weeks)

- Use the algorithms and the store data methods to make methods connected to the main GUI

Test Plan



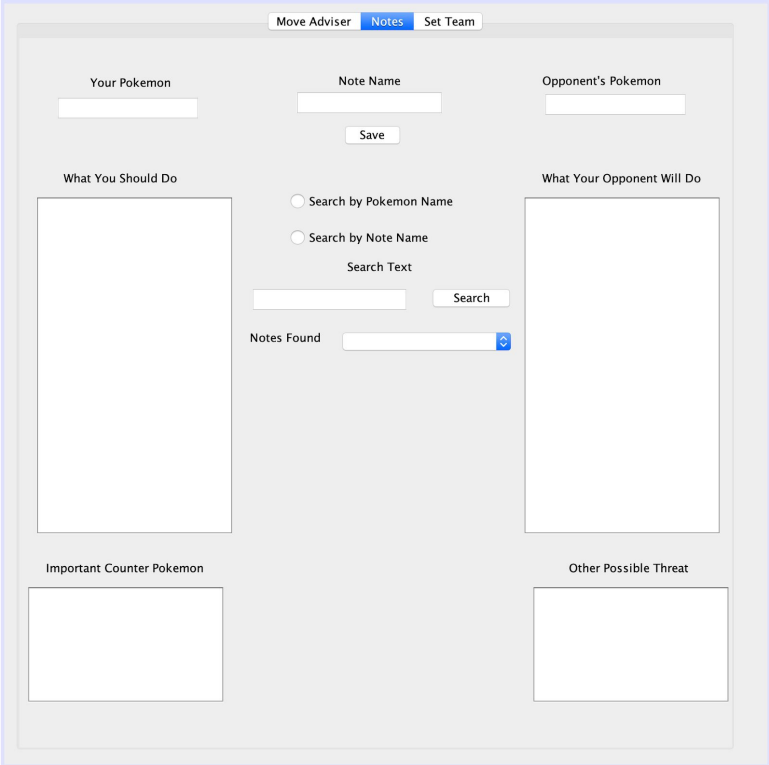
Move Advisor Tab:

Things done on this tab:

- Internal Functioning
    - Calculate and predict the opponent's set
    - Calculate Weakness
    - Process data and give advice/predict
    - Record data

Test Type	Nature of Test	Example
Advice and predicts are as expected from the algorithm	See whether the advice/predict are according to the algorithm	
Test with simple data	Check whether the program runs	Use inputs of correct formats
Test with inappropriate data	Check program does not run	Wrong pokemon name/move
Test record function	Rerun the program with a recorded set and check whether it predicts using recorded data	

Notes

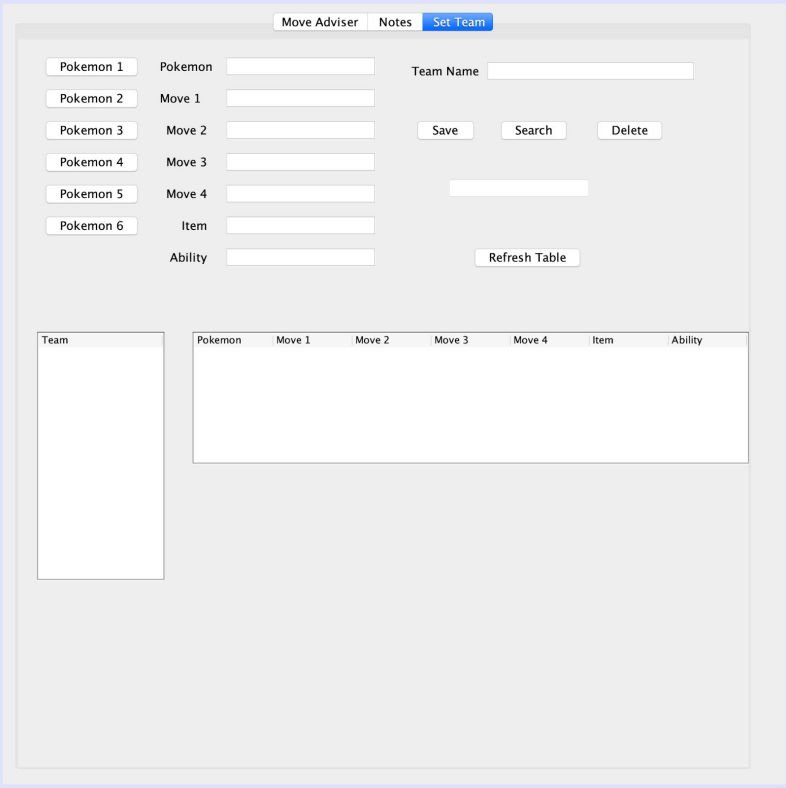


Things done on this tab:

- Inputs and Outputs
  - Pokemons
  - Written notes

Test Type	Nature of Test	Example
Test record with simple data	Check whether the program record notes	Use inputs of correct formats
Test with inappropriate data	Check program does not run	Did not insert advice note
Test view record and searches	Check whether saved notes can be searched and viewed	Search “Clefable”

Set Team Page



Things done on this tab:

- Inputs and Outputs
  - Pokemon Teams
  - Pokemon Sets

Test Type	Nature of Test	Example
Test record with simple data	Check whether the program record notes	Use inputs of correct formats
Test with inappropriate data	Check program does not run	Enter nonexistence move
Test view team and searches	Check whether saved notes can be searched and viewed	Search “HyperOffensive1” If search not found, display error text

Word Count: 252