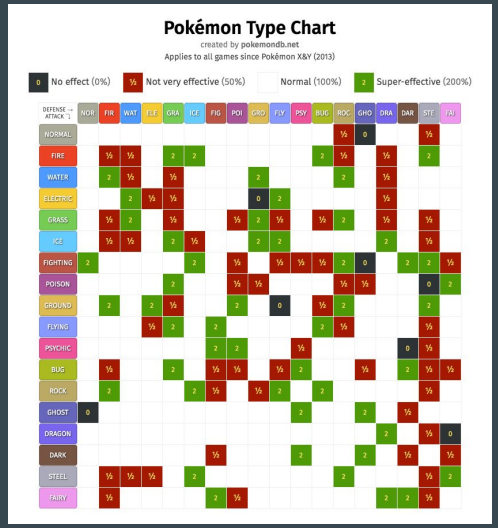
Pokemon Team Builder

Group 114-4
Ben, Taryn, Jamie, Christian, Trustin, and Kaleb

What is Pokemon?-Trustin

- Can be treated as a card game (YuGiOh, Hearthstone)
- Tournaments where people build the most optimal team to win
- Rock, paper, scissor mechanics in pokemon types
- Tons of math calculating damage, defense etc.



Formula for Catching Pokemon

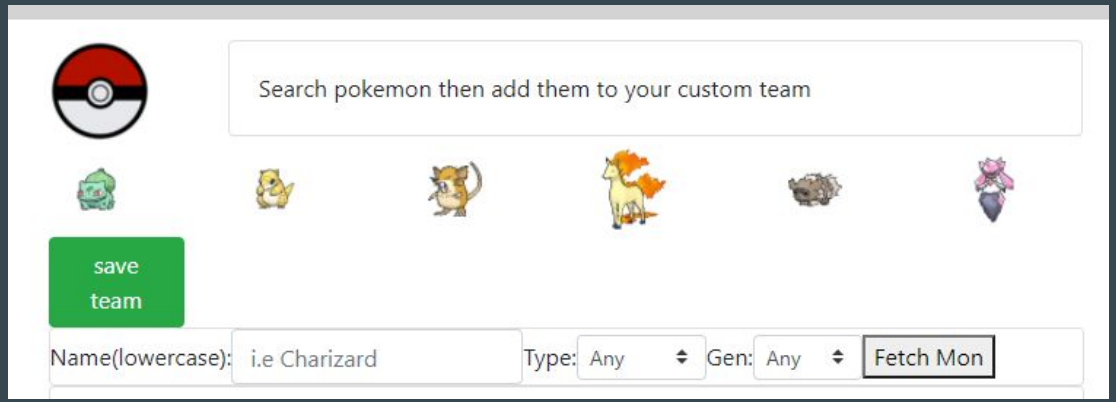
$$a = \frac{(3 \times \mathrm{HP}_{\max} - 2 \times \mathrm{HP}_{\mathrm{current}}) \times \mathrm{rate} \times \mathrm{bonus}_{\mathrm{ball}}}{3 \times \mathrm{HP}_{\max}} \times \mathrm{bonus}_{\mathrm{status}}$$

What I Think the Formula is

$$a = \text{Number of times I press A}$$

Project Description-Christian

- Pokemon Team Builder
- Competitive pokemon fighting
- Plan out potential teams
- Customize Stats and pokemon on team
- User profiles to store teams



Tool Rating- Taryn

Jira - project tracker
★★☆☆☆
Github - version control
★★★★☆
PostgreSQL - Database
★★★★★
Integrative Testing
★★★☆☆
Heroku - deployment/hosting
★★★★★
VSCode - IDE
★★★★★
Node.js - server
★★★★★

Challenges

Problem 1: Teamwork- Ben

- Waterfall didn't allow too much for communication

Solution:

- Eventually Waterfall was too slow so we had to really communicate and use a bit of agile

Problem 2: Injury- Kaleb

- Kaleb injured his hand so more workload was put onto Ben in the back end

Solution:

- Increased communication helped

Problem 3: C++-Kaleb

- Initially we were going to use C++ for a node addon to do calculations but implementing that ended up being too complicated.

Solution:

- Did all calculations in javascript

HTML - Challenges

- Converting the standard HTML templates into EJS-Trustin

- Responsive EV Sliders with limited EVs- Christian

- Timelines and Waiting- Trustin

PostgreSQL  - Jamie and Taryn

- Database management tool of choice
- Pokemon Table
  - Includes all Pokemon from the franchise, as well as their max stats
  - Imported 800+ Pokemon from a .csv and inserted values into the table
- Users
  - Creates and stores new users for registration and login
- Teams
  - For each user, stores teams of Pokemon with custom stats
- Accessed using Heroku Postgres database service

PostgreSQL - Challenges  - Jamie and Taryn

- Initial Set-Up
  - Teaching ourselves about relational database management systems prior to labs
  - Setting up a server
    - Localhost -> Heroku
- User Authentication
  - Variety of ways to set up a user registration system
    - Passport, Node Express, etc
  - Inserting values into the database
    - Taking input values from HTML/EJS files and inserting them into the database
- Safety Concerns
  - Non-encrypted
  - Implementing third-party authentication solutions
    - Google Identity Platform

NodeJS/npm - with express and pg promise Ben

Node js = 5/5 Used to build the server and make pages dynamic with ejs

Express package = 5/5 Handled routing/request and caching session data

Body parser package = 5/5 Handled sending and receiving json data between server and client

Pg promise = 5/5 Handled database connection and modification.

NodeJS Challenges Ben

- Learning node JS and how it interacts with all of the different tools used.

- Making each clients session unique.
- Dependent on the format of other tools *(e.g.,* structure of database and html pages).
- Constructing dynamic queries/inputs to handle of varying sizes and data.
- Handling data transfer of server -> client and server -> postgres of the saved teams of pokemon.

Heroku - Ben

App Deployment - Heroku allowed for easy deployment to the web. Heroku can identify a Node js app and deploy it by creating a heroku git.

NPM - Heroku will take your package.json and deploy node js app with all the needed add ons.

No challenges but our scale is limited by the free heroku.

Jira  - Christian

- Agile project management tool
- Tracks completed and in progress features
- Used to manage project development
- Sprints managed features in development

Jira - Challenges  - Kaleb

- Checking Jira - It was easier to communicate through discord so jira wasn't checked as frequently as it should have been
- Learning how to use jira
  - Tool was new to all of us
  - Agile was new to all of us

Extra slide - Kaleb

Before I busted my hand I was supposed to:

- Implement functions to show the attack and defense Multipliers for each pokemon type. (Functions written just not implemented into web page)
- Show the types that a team would be effective against using type data from each pokemon and show it on team page/team builder

```
function effectivity(type1, type2) {
    var types = ["normal", "fighting", "flying", "poison", "ground", "rock", "bug", "ghost", "steel", "fire",
        "water", "grass", "electric", "psychic", "ice", "dragon", "dark", "fairy", "none"];
    var t1, t2 = 0;

    for (var i = 0; i < 18; i++) {
        if (type1 == types[i]) t1 = i;
        if (type2 == types[i]) t2 = i;
    }
    var effective_array =  [[1, 1, 1, 1, 1, 0.5, 1, 0, 0.5, 1, 1, 1, 1, 1, 1, 1, 1],
                            [2, 1, 0.5, 0.5, 1, 2, 0.5, 0, 2, 1, 1, 1, 1, 0.5, 2, 1, 2, 0.5],
                            [1, 2, 1, 1, 1, 0.5, 2, 1, 0.5, 1, 1, 2, 0.5, 1, 1, 1, 1, 1],
                            [1, 1, 1, 0.5, 0.5, 0.5, 1, 0.5, 0, 1, 1, 2, 1, 1, 1, 1, 1, 2],
                            [1, 1, 0, 2, 1, 2, 0.5, 1, 2, 2, 1, 0.5, 2, 1, 1, 1, 1, 1],
                            [1, 0.5, 2, 1, 0.5, 1, 2, 1, 0.5, 2, 1, 1, 1, 1, 2, 1, 1, 1],
                            [1, 0.5, 0.5, 0.5, 1, 1, 1, 1, 0.5, 0.5, 0.5, 1, 2, 1, 2, 1, 1, 2, 0.5],
                            [0, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 1, 1, 0.5, 1],
                            [1, 1, 1, 1, 1, 2, 1, 1, 0.5, 0.5, 0.5, 1, 0.5, 1, 2, 1, 1, 2],
                            [1, 1, 1, 1, 1, 0.5, 2, 1, 2, 0.5, 0.5, 2, 1, 1, 2, 0.5, 1, 1],
                            [1, 1, 1, 1, 2, 2, 1, 1, 1, 2, 0.5, 0.5, 1, 1, 1, 0.5, 1, 1],
                            [1, 1, 0.5, 0.5, 2, 2, 0.5, 1, 0.5, 0.5, 2, 0.5, 1, 1, 1, 0.5, 1, 1],
                            [1, 1, 2, 1, 0, 1, 1, 1, 1, 1, 2, 0.5, 0.5, 1, 1, 0.5, 1, 1],
                            [1, 2, 1, 2, 1, 1, 1, 1, 0.5, 1, 1, 1, 1, 0.5, 1, 1, 0, 1],
                            [1, 1, 2, 1, 2, 1, 1, 1, 0.5, 0.5, 0.5, 2, 1, 1, 0.5, 2, 1, 1],
                            [1, 1, 1, 1, 1, 1, 1, 1, 0.5, 1, 1, 1, 1, 1, 1, 2, 1, 0],
                            [1, 0.5, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 1, 1, 0.5, 0.5],
                            [1, 2, 1, 0.5, 1, 1, 1, 1, 0.5, 0.5, 1, 1, 1, 1, 1, 2, 2, 1]]
    var return_arr = [];
    for(var i = 0; i < 18; i++)
    {
        var value = effective_array[t1][i] * effective_array[i][t2];
        return_arr.push(value.toString());
    }
    return return_arr;


}
```

```
me@DESKTOP-0PK0ECA:/mnt/c/Users/kjp19/Documents/GitHub/114-4/Calculations$ node effectivity.js
poison,fighting receives damage from:
normal 1
fighting 1
flying 2
poison 0.5
ground 0.5
rock 0.25
bug 0.5
ghost 0.5
steel 0
fire 1
water 1
grass 2
electric 1
psychic 2
ice 1
dragon 1
dark 0.5
fairy 4


poison does damage to:
normal 1
fighting 1
flying 1
poison 0.5
ground 0.5
rock 0.5
bug 1
ghost 0.5
steel 0
fire 1
water 1
grass 2
electric 1
psychic 1
ice 1
dragon 1
dark 1
fairy 2
```

Demo time!