Benjamin Roseburrough

Stevene Alvarez

Travis Tharp

Nonelaci Her

Derek Greeley

Final Project Proposal

First Idea: Using Machine Learning to find a correlation between character parameters and overall match-up scores to predict the overall match-up score of a newly upcoming character.

~~Backup Idea: Do Fantasy Football analysis. For Fantasy Drafts, find out what players get you the best value per dollar.~~

~~Backup Idea: Housing analysis.~~

Core Message/Hypothesis: How well does the correlation between character parameters and overall match-up scores help predict the overall match-up score of a newly upcoming character Min Min (and the other 5 DLC characters)?

Questions:

* What are the character parameters that best predict a character’s performance as measured by the overall match-up score?
  + The overall character match-up score is the total of all individual match-up scores that a character has against all other characters (ex. Lucina vs. Fox, Link, Snake, etc.).
* Which of the DLC characters is the best and why?
* If certain traits are significantly more important than others and top characters consistently fit a certain archetype (based on stats), what can developers learn based on our model to better balance the game and ensure that no one type of character is too dominant?
  + As an example, if frame data is by far the best predictor of match-up scores, then would it be sensible to make fast characters deficient in most or all other parameters?

Sources of Data:

* <https://github.com/Frannsoft/FrannHammer> - For character parameter data (X values).
* <https://docs.google.com/spreadsheets/d/1QV2_WC--SEPUVM5U2qvgh_uJHKPhZzXAcDhDE6irtP4/edit#gid=1934011249> – For character match-up data.

Data Calling (SQL): (Derek, Stevene, Travis)

* Open Postgres/pg Admin4.

Character Parameters

* Right click on ‘Servers’ to the left.
* Specify the URL for the character attribute API from GitHub.
* Use postgres for both the username and password?
* Check to make sure that a table was created for the character table in question.
  + There is a table for character data, move data, movement data, character attribute data, and throw data.
  + Character data includes:
    - OwnerID (Character ID)
    - DisplayName (Name of character as used on the site)
  + Move data includes:
    - Name (Category of attack – jab, tilt, etc.)
    - OwnerID (Character ID)
    - Owner (Name of character)
    - HitboxActive (Either when attack first becomes visible?)
    - FirstActionableFrame (When the attack has finally bypassed the startup frames and can actually hit enemies or otherwise become active)
    - BaseDamage (How much damage the attack does in terms of % before any buffs or penalties apply)
    - Angle (The angle that the attacks sets the foe flying at)
    - BaseKnockBackSetKnockback (The amount of knockback that an attack has)
  + Movement data includes:
    - Weight (How heavy/knockback resistant the character is)
    - Max Jumps (How many jumps a character can use before getting to the ground)
    - Ability to tether (Grab onto the side of the stage)
    - Wall Jumps (Ability to use the side of a stage to gain an extra jump)
    - Wall Cling (Ability to cling onto the side of a stage)
  + Character Attributes data includes:
    - Gravity
    - Fall Speed (Max Fall, Fast Fall, Regular Fall)
    - Run Speed
  + Throw Data

Character Match-Ups

* Right click on ‘Databases’ under ‘PostgreSQL 11’ on the left side of the screen.
* Select ‘Create’ and then ‘Database’.
* Create a name such as ‘Character\_Match-Ups’.
* When the database is created, run a query with ‘CREATE TABLE’ to create a table with columns that match the match-up chart found on <https://docs.google.com/spreadsheets/d/1QV2_WC--SEPUVM5U2qvgh_uJHKPhZzXAcDhDE6irtP4/edit#gid=1934011249>.
* Download that CSV file from the site onto your computer.
* Right click on the ‘Tables’ option under ‘Schemas’ and select ‘Import/Export’
* Import the CSV file from the site.
* Create a column that adds up values across rows.

Data Joining (Python): (Derek, Stevene, Travis)

* Join the data from the character, move, movement, attribute, and throw tables with ‘join’ statements for a character.
* Ensure that all data can be condensed into a minimal amount of rows (preferably one).
* Repeat for all other characters.
  + Use the ‘append’ function to push the data for each character into one central table.
  + Ex. Push the data for Marth onto a table with Bowser’s data, then push the data for Fox onto the table with to Bowser’s and Marth’s data.
* If there are far too many character features, consider dropping some of them based on a group discussion.

Machine Learning (Python): (Ben)

* Run Multiple Regression Analysis to create a model that will make predictions of character match-up scores.
* Determine the combination of features that generates the optimal model.
* Rerun the analysis to create an optimal model.
* Train the model on data from all characters from the initial release of the game.
* Use the model to make predictions about the overall match-up scores of the DLC characters.
* Save predictions on local computer.
* Push to GitHub.

Presentation (Tableau): (Laci)

* Import the data from the master table into Tableau.
* Import the predictions into Tableau.
* Create a histogram with data from the initial release characters.
* Overlay with a histogram containing data from the DLC characters.