COMPSCI 381-01 – JavaScript and DHTML – Spring 2025

Plants Vs. Zombies Heroes Evolutionary Leap Query Database

Andy Brandt (BrandtA13@uww.edu)

1. Introduction

Plants vs. Zombies Heroes is a digital collectible card game developed by PopCap Games and published by Electronic Arts. Released in October 2016 for iOS and Android, it is a spin-off of the popular *Plants vs. Zombies* franchise. Unlike the tower defense style of the original series, *PvZ Heroes* features a turn-based card game format, where players can choose to play as either plant or zombie heroes, each with unique abilities. But due to the gameplay being wildly different from other installations in the franchise, many players were put off from the game. This quickly resulted in many players feeling dissatisfied with the game, and updates quickly dwindled in the coming years. The last expansion, *Triassic Triumph*, was published in 2018, and the last patch balancing cards to keep strategies in equal powers of level was in 2019. However, instead of deleting the game, the remaining players continued to play *PvZ Heroes*, many continuing to this day (me included). The meta (strategies and decks that players would run to consistently win games) continued to progress, with players finding what the de facto best strategy is, and which decks can be used to best counter the meta, creating a new meta in its place. Then, in preparation for *Plants vs. Zombies* 16th birthday, *Heroes* ended up getting a surprise balance patch on December 9th, 2024. No one was expecting an update to the game, and even though the update only brought some changes to units' strength and health, it was a welcome change compared to the last 5 years of updates (being none).

Combat in the game is quite simple, the game starts with each player drawing four cards from their deck, then getting a random superpower card based on what hero they decided to be the leader of their deck. Then, both players get 1 sun or brain (based on if they chose to play as plants or zombies respectively) to play the cards in their hand. First, the zombie player gets to play zombies cards in one of five lanes, then plants respond by playing any card in their hand (up to the amount sun they have for the round). Finally, zombies get to counter what plants play with a phase where they can play only tricks (one-time abilities that can boost the stats of a card, or deal damage and remove cards from the board) and environments (cards that add modifiers to cards that are played inside). The round concludes with combat, where each plant and zombies attacks each other, dealing its attack (number in the green circle) to an opposing unit's health (number in the red heart). If a card's health reaches zero, the card is removed from the game. If there is no unit to block the damage, the damage is subtracted from your hero's health, and if your hero's health reaches 0, you lose. In the image above, the plant hero would take 5 damage from the left zombie. The first pepper would die from the pirate zombie, who then loses 2 health from the pepper attacking back. Then the zombie hero would take 6 damage from the pear and the second pepper.

Many card games' metas result in the best deck in the room having a clear way of winning games, and a consistent way to achieve victory. In contrast, *Heroes* has an astonishing number of abilities that completely rely on RNG. Its meta is no different with the best decks trying to minimize the RNG the game loves to use. Instead, my favorite strategy leans into the random nature of *Heroes*. The deck relies using a card called Evolutionary Leap. This card turns one zombie unit into a random zombie unit that costs one more than what it was used on. For example, Evolutionary Leap was used on a 3-cost card, the zombie it transforms into to would be a random zombie that costs 4 brains. The deck shown on the left likes to Leap cards that have strong abilities that occur when played and transform them into strong bodies that the opponent can't deal with.

In the 7 years that I've been playing *Plants vs. Zombies Heroes*, I have never found a service that could help me predict what card I could expect to receive. I like to use the card very defensively, as when a card is transformed, the card that replaces it starts with full health. For example, if the plant player plays a Three-Headed Chomper in front of my Ra Zombie, I can expect to lose not only Ra Zombie when in combat begins but also lose the zombies in the two adjacent lanes. I could choose to play an Evolutionary Leap on my Ra Zombie, but I'm not sure what 6-zombies could be

used to deal with the Three-Headed Chomper. I want to be sure that the "leap pool" has cards that can eliminate this difficult threat.

2. Project Goals

At minimum, I would like to create a database of every card in *PvZ Heroes*, and have a function that would filter through the database for all the units of the selected team (as those are the only cards that can be "Leaped"), then filter again based on three restrictions: an exact cost of a card, a minimum strength stat, and a minimum health stat.

As for stretch goals, the first should be to add a percentage of cards that meet one of the three requirements. If I'm looking to Leap my Ra Zombie to deal with a Three-Headed Chomper, I want to see that my leap would result in a positive outcome only 38% of the time.

Another goal could be to create a checkbox that will let me search with a card's effective health. Some cards gain strength and health when they are played, and this effect still triggers when they are Leaped into. One of these cards is the Deep Sea Gargantuar, which has an odd description, but

can be read as “When Played: this gets +1/+1. When a zombie moves to a new lane (including when playing a zombie) that zombie also gets +1/+1.”

With this added, you should be able to see that our positive outcome jumps from ~38% to ~53%! Fun fact: with this function created, you’re also able to see that the only 6-cost card that can not only deal enough damage to destroy the Three-Headed Chomper but also survive the fight.

My final stretch goal is to add a visual list to show what cards have been queried. This will be the hardest to create, as I had to find images of every card online, and add them to each card object. A few of the cards still have images from before the balance patch in 2024, so some cards may seem like they should not fit in a given query. Here I have shown one of the most egregious offenders, the Prickly Pear. Above is the image I could source to use in my website and below is its stats after its rework in 2025. Additionally, when used with the effective strength/health query, the new cards will show up in the visual list. These will seem out of place as well, as their images show lower stats than what the query is asking for.

3. Design and Implementation

Before constructing the website, I decided to create the object arrays for not only the plant and zombie cards, but also the 22 heroes and their signature superpowers. I wasn't sure how large I wanted to expand the database in the future so I thought it would be a good idea to start large rather than modify later. I decided to have these be separate JS files as the had little to do with the programming portion, and they were hundreds of lines long. They are accessed the same way you access the main JS program.

The first function I created was the updateCardVisual function, which takes an array of card objects and adds their images onto the website. It first removes the innerHTML from the previous query, then appends each card to a div element with the createCardVisual function. This function takes one card object and returns a div element a class to create a flex box with the card's rarity's color, and innerHTML for the image.

The rest of the program is inside a function tied to the query button. First, we find if we are querying the plant card or the zombie cards (we can grab this from the Leap Plants/Zombies radio). Then we copy that team's list to the unitList array, and filter for only the unit cards (the only type of card that can be Leaped).

Then, I check if either of the effective Strength/Health checkboxes are checked. If they are, I change the cards strength/health to be what it's stats would look like in the real game. If one is not checked, I rewrite the card's strength/health variable with what the originally were. This is done with the 'effectiveStrength' and 'ogStrength' variables inside card objects that involve this (the health variable are named the same way). I feel like this is the snippet of code is conducted the most poor, but it does make the segment much easier to execute.

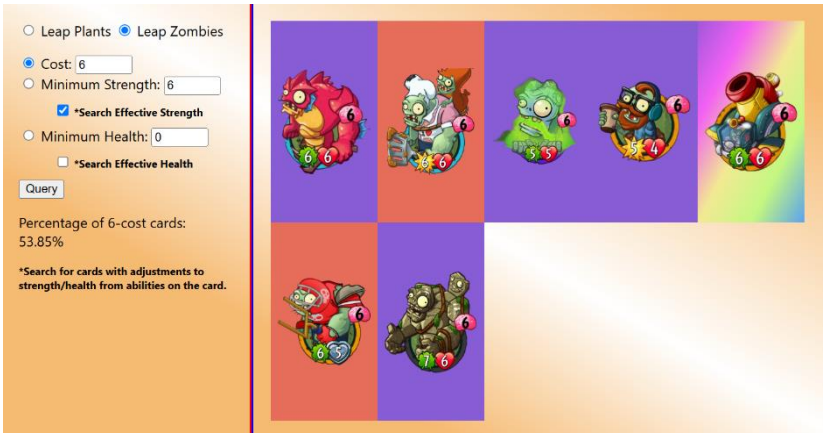
And finally, we can query the three restrictions: card cost, minimum strength, and minimum health. With how I created the effective stat snippet, this can be done quite efficiently, as we don't need to create 4 different copies, searching between strength, health, effective strength, and effective health. Once through this, we sent the queried list to the updateCardVisual function.

The last portion of the program counts how many of the cards in the queried list meet the full list we are comparing it to. Depending on which radio is checked, it will create innerText and add it to a p element, which ends up showing the percentage of cards that meet just one of the queries.

When creating this program, I decided to weaponize the .class selector in the CSS file. This helped a lot in creating the boxes for the image chain. If I did not use the .class selectors, I would have had to create 7 separate card containers with their only difference being the background color.

4. Testing the Program

Overall, there is not much to teach on how to operate this program. Much of this is self-explanatory, but a few of the radio placement are difficult to associate with each other.



First, choose whether you want to query the plant cards or the zombie cards. This is done with the first radio on the top row.The next radio goes across the next 3 lines, between Cost, Minimum Strength, and Minimum Health. It chooses what the 'master query' is, its used for calculating the percentage of total queried card that meet only one of the conditions. Separate from this are the 2

checkboxes, asking if you want to search with the card's strength/health that is would reasonably have inside an actual game.

Once you hit 'Query', the program will execute, and create a list of images on the right that represent the cards that meet your query.