California University of Pennsylvania

CSC 490: Senior Project

Project Vulpix

Specification Document

Andrew Siddall

Adlene Bellaoucha

Mathew Bedillion

Christopher Crisson

November 13, 2018

Instructor Comments/Evaluation

Table of Contents

Abstract

Project Vulpix aims to create software to help players of the Pokémon Trading Card Game. Our motivation for this product comes from the high investment currently required to become competitive at the game. Newer players are especially turned off by the high costs associated. The software will provide a competitive Artificial Intelligence to help players improve their skills. We will provide an interface to simulate and analyze the game itself. This document intends to provide both the client and development team with the requirements and functionality expected of the final product.

Document Description

Purpose and Use

This document is intended to define the scope of the product and the specific requirements the development team will be held to. In this interest, the development team hopes to provide sufficient detail as to what the product will do. Upon acceptance of the terms put forth in this document by both the development team and the client, this document will serve as a contract binding both parties.

Intended Audience

This document has been drafted to the best of the development team's ability to be neither arduous in technical detail or uselessly simplistic. It is encouraged that readers to seek clarification on any and all sections of the document that are ambiguous, unclear, or concerning in any way. This document will describe the features and costs associated with the production of the software it describes. This document will also be used by the development team to enforce proper direction and priorities.

System Description

Overview

The software this document describes will be a trainer for the Pokémon Trading card game. It will provide an interface for a user to describe the “state” of Pokémon trading card game. It will provide the user with what it determines to be the action that most likely leads to winning the game. The interface will be intuitive to users familiar with the game.

Environment and Constraints

End User Profile

The End User for Project Vulpix will be players of the Pokémon Trading card game. The software in not intended for, and will be of no use to, a user unfamiliar with the basic rules of the game. Beyond that prerequisite, anyone will be a prospective end user for the software. The software will only be available in English. Any localization of the software is beyond the scope of this document and is unfeasible due to time constraints. The primary target audience will be players that would like to improve their gameplay abilities.

User Interaction

The user will interact with Project Vulpix in two basic ways. The main way the software will be used is through entering game state information. The user will “describe” each “game zone” to the software. This will be done by first choosing the “deck” each player is using. Then selecting cards from the decks to place into these “game zones”, effectively creating a model of a match of the Pokémon trading card game. Once this information is entered, the software will analyze the information and provide what it determines to be the best game action for the player to perform. The user will then enter new information reflecting the game action and any actions performed by the other player and rerun the analysis to get another suggested game action.

The other way users will be able to interact with the software will be by entering “deck lists”. These are lists of cards describing the cards each player uses to play the game. The software will have some decks premade for users to use “out of the box”.

Hardware Constraints

The hardware constraints of Project Vulpix are not able to be fully described with certainty, so a very conservative approach is being taken to describe what is expected. The software is expected to require a fair amount of free space on disc (up to 2 GB), a reasonable amount of RAM (8 GB), and a reasonably new multicore processor (likely an intel i3 equivalent released after 2015). A desktop or laptop computer that the average college student might own should be sufficient for reasonable runtimes.

The hardware constraints above are intended to apply to achieving reasonable runtimes. This product is not part of a real-time system and will thus not guarantee minimum runtimes. In fact, the means that will be employed to achieve the desired results of the program will be built with ability to halt the current analysis and report the current best action to the user.

Software Constraints

The software will run on Windows 10. The basic software constraint will be that the target system must be able to run Python 3, which is released as open source software (General Python FAQ; What is the Python Software Foundation). There will be several python packages that will be used in the project. Only open source packages will be included in the final delivered source code.

Time Constraints

The deadline for this project will be determined by the client. Because this is an academic project, the time table will be based on the length of the semester. Expected milestones will be determined with respect to this constraint. The development team expects many hours to go into the active development of the project. The team's expectation to be in many of the same classes should facilitate consistent communication and motivation to complete the project within the agreed upon time.

Cost Constraints

Project Vulpix will be released as open source software ("What is open source?"). It will be available to any person at no cost. Any persons attempting to charge for any portion of the product will be in violation of the end user agreement.

The software will consist partially of components made available freely by persons not related to the development team. All the components will have been released as open source software.

The cost constraint to the user will be that of providing a system that meets the system requirements for installing the software on the system. The expected price of a new machine capable of running the software is around five hundred USD at the time of writing. We provide this estimate as a guideline only. We cannot be held responsible if this estimate is deemed inaccurate due to our inability to control prices or availability of products that are not the software described in this document.

Acceptance Test Criteria

Testers

The development team will be the main testers on the cases. This testing will go through all parties to be sure that nothing is missed. Testing will be completed as we continue implementing new features. With the resources each of us holds, this will give us the best possibility of finding all problems and getting them fixed before getting too far. With each member having their own ways of doing things, it is best to have everyone rest before continuing to the next step.

Criteria for User Acceptance

Project Vulpix will be considered successful if the client's parameters have been met. As well as each of the following functions are required:

- The final product can be installed on any machine that meets the base requirements of the program.
- The user can start a new game.
- The user can set the beginning game state.

- The user can update their cards in hand.
- The user can update the cards in play.
- The user can submit the game state for analysis
- The program will output to the user what it determines to be the optimal game action.
- The user can edit and create decks.
- All correct data entered is accepted, and incorrect data is met with an error message.
- The user interface will be direct, user friendly, and easily operated.

Integration of Separate Parts and Installation

The software will be deliverable as an executable file. The installation process will be simple, menu driven, and familiar to people with basic computer skills. It will be installed to the user's local disk. Once the program is installed and run it will provide the user with the graphical user interface.

Hardware Requirements

- A computer system capable of running a modern operating system
- ~ 2 GB of internal storage
- 8 GB of RAM
- Intel i3 processor or AMD equivalent
- Keyboard
- Mouse
- Display

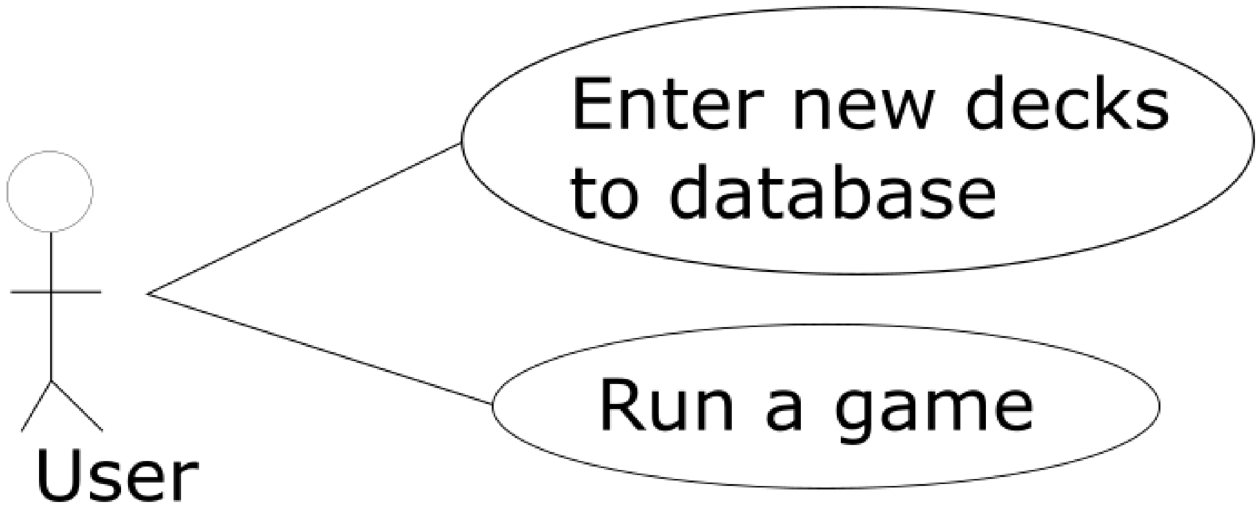
Software Requirements

- Windows 10
- Python 3

System Modeling

Functional: Use Cases and Scenarios

The following use case will show the two choices the user will choose between when first starting the program. The following Design workflow will describe the more specific user interactions with the software.



This initial use case shows the first actions the user can take when launching the program. "Enter new decks to database" allows the user to access the deck lists available for setting the game state. "Run a game" will bring the user begin setting the game state.

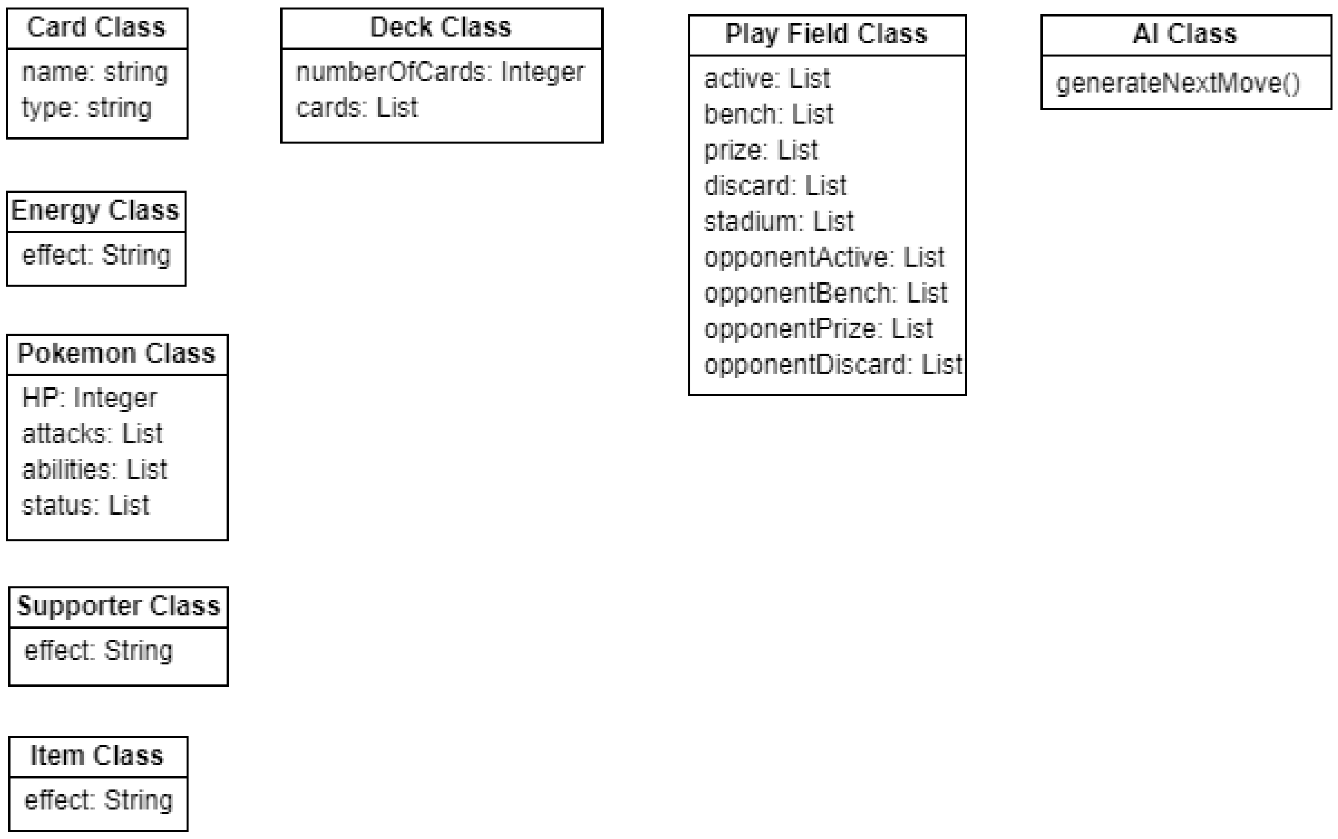
Normal Scenario

1. The user launches the program.
2. The user chooses to run a game.
3. The user chooses two decks to play against each other.
4. The user sets the game state.
5. The game state is validated.
6. The user starts the analysis.
7. The user is provided with the optimal game action.
8. The user adjusts the game state.
9. Points seven and eight are repeated until a win condition is reached.
10. The user is returned to the start screen.

Exception Scenario

1. The user launches the program.
2. The user chooses to run a game.
3. The user chooses two decks to play against each other.
4. The user sets the game state.
5. The game state is determined to be invalid.
6. The user is prompted with the error.
7. The user adjusts the game state to correct the error.

Entity: Class Diagram



Class Descriptions

Card Class

The card class is the base class that all cards will inherit from. All cards will have a name and type.

Energy Class

The Energy class will inherit from the Card class. It will have one additional attribute that will describe what kind of energy effect it has.

Pokémon Class

The Pokémon class will inherit from the Card class. It will have four additional attributes. These will describe the amount of health points (HP) it has, the attacks, abilities and any status aliment the card has accrued.

Supporter Class

The Supporter class will inherit from the Card class. It will have one additional attribute that will describe what kind of supporter effect it has.

Item Class

The Item class will inherit from the Card class. It will have one additional attribute that will describe what kind of item effect it has.

Deck Class

The Deck class will use an integer to keep track of the number of cards it contains. All of these cards will be stored in a list.

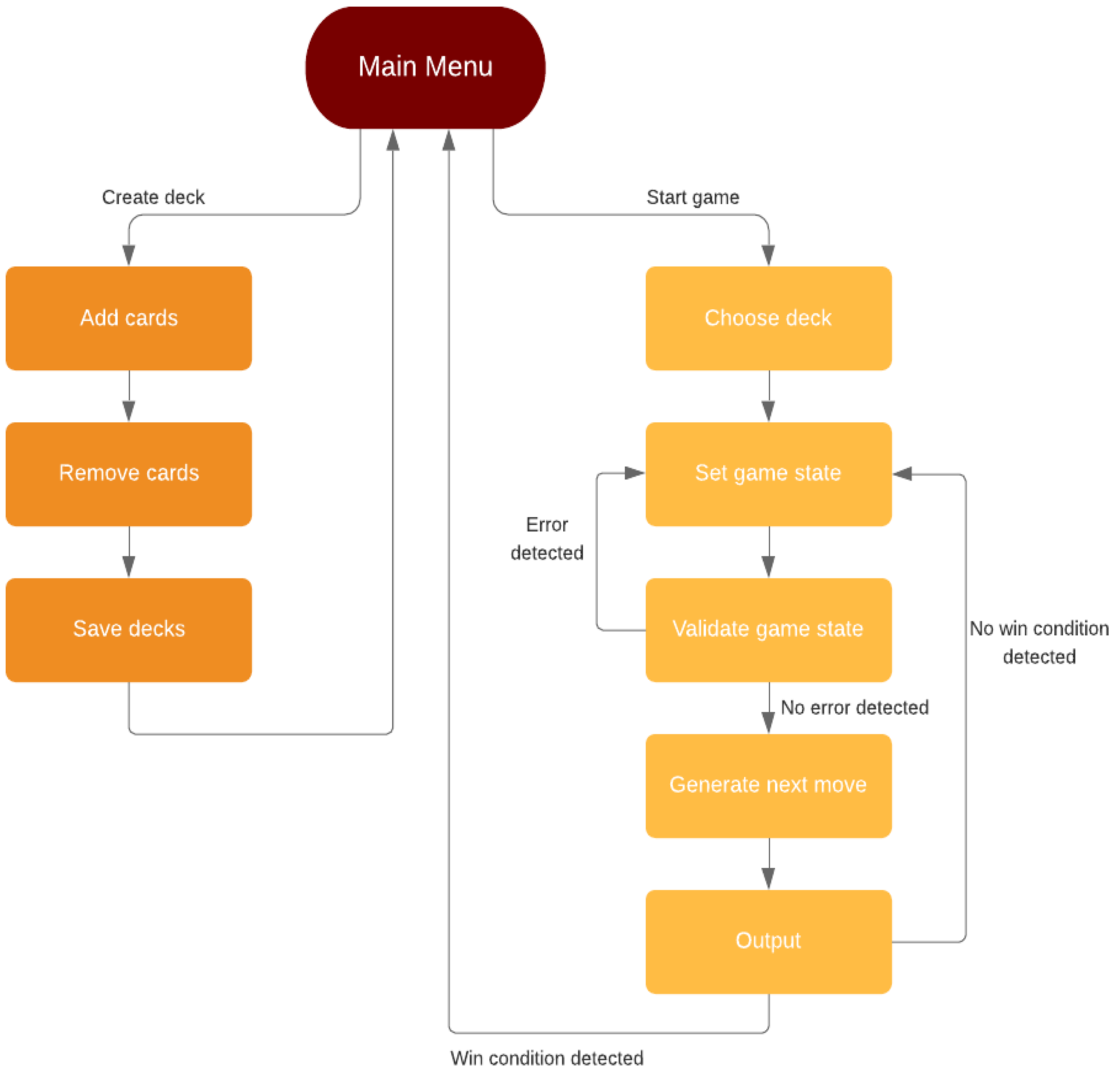
Playfield Class

The playfield class will describe the game state. This includes all game zones controlled by both the user and the opponent.

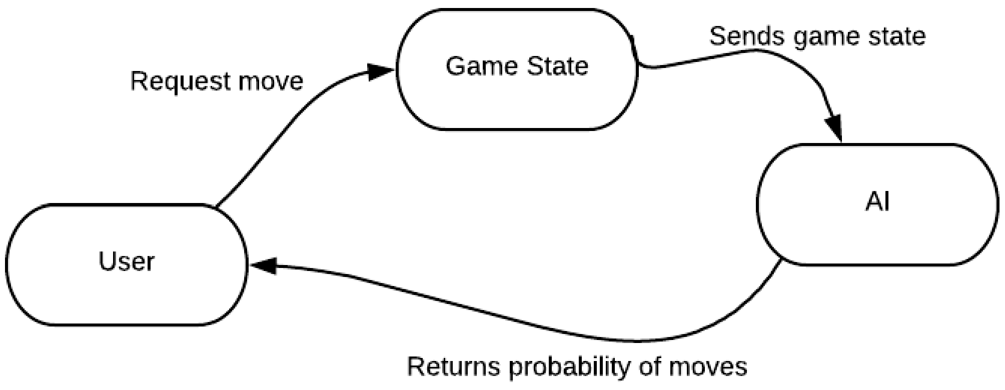
AI Class

The AI class represents the algorithm that the program will use to determine the optimal game action for a user to take.

Dynamic: Statechart



Dataflow Diagram



Components needed for Implementation

Hardware

- Computer system Windows 10 operating system
- Keyboard
- Mouse
- Monitor

Software

- Python 3 interpreter
- Text editor
- Several Python 3 libraries: TensorFlow, PySci, etc...

Appendix I: How to play: Pokémon Trading Card Game

The Pokémon Trading card game is a card game consisting of two players with sixty card decks. The players take turn and can only perform actions on their turn. At the start of the game, each player shuffles their deck and draws seven cards. Each player looks at their hand and determines if they have a basic Pokémon. If they have a basic Pokémon and it is their only Pokémon, they put it face down in the active position. If there are more than one basic Pokémon in the player's hand, they choose one of their basic Pokémon and places it in the active position. They also have the option to place the other basic Pokémon on the bench if they wish. If no basic Pokémon are found in the opening hand, then that player takes a mulligan. A mulligan is when the player reshuffles their hand into their deck and tries again; This happens until a basic Pokémon is found. When a player has a mulligan, the opposing player can choose to draw an extra card for each mulligan the player takes. Once both players have their basic Pokémon face down in the active position, the players layout six prize cards. After the active Pokémon and prizes are placed, the players flip a coin to determine who goes first, both players flip their Pokémon in play, and the game begins.

At the beginning of each player's turn, they draw a card. During each player's turn, they can perform several actions based on what cards are in their hands such as: play a supporter, play an energy, play item cards, play new basic Pokémon, evolve Pokémon, use abilities, and attack. A player may only play one supporter and/or energy card from their hand per turn. If the player has an open spot on their bench, typically less than six Pokémon on the bench, then they may play basic Pokémon from their hand until their bench is full. Pokémon who evolve from a Pokémon on the bench can be placed onto that Pokémon, known as evolving a Pokémon. As many item cards as the player has in their hand can be played per turn. Some Pokémon have special abilities that can be triggered during a player's turn. Each of these abilities has limits of the number of times they can be done per turn, specified on the card itself. The turn ends when a player signals to their opponent that their turn is done, passes, or when the player decides to attack the opposing Pokémon. To attack, the player's active Pokémon must have the correct type and amount of energy required to do the attack. When an attack is used, the appropriate action is taken based on what the card says the attack does. Typically, this involves placing damage counters on the defending Pokémon. If the damage counter from the attack is greater than the defending Pokémon's hit points, then the defending Pokémon is knocked out and the player takes one of his face down prizes.

The game has three-win conditions: Deck out, knocking out all the opponent's Pokémon, and taking all your prizes. A deck out is when the opponent has no more cards in their deck and is unable to draw for their turn. When the active Pokémon is knocked out, the player whose Pokémon was knocked out chooses one of the bench Pokémon to promote into the active. If there are no Pokémon on the bench to promote into the active, that player loses. Each time a player knocks out a Pokémon, that player gets to take one of their face down prize cards. If there are no more prize cards left after the player takes theirs, then that player wins the game.

Appendix II: Technical Glossary

Ability: An Ability is an effect on a Pokémon that is not an attack. Some will be active all of the time, while some you will need to choose to use. Read each Ability to make sure you understand exactly how and when it works.

Active Pokémon: The player's in-play Pokémon that is not on the Bench. Only the Active Pokémon can attack.

Attach: When you take a card from your hand and put it on one of your Pokémon in play.

Attack: 1) When your Active Pokémon fights your opponent's Pokémon. 2) The text written on each Pokémon card that shows what it does when it attacks (a Pokémon can have several attacks on it).

Attacking Pokémon: The Active Pokémon, as it performs an attack.

Basic Energy Card: A Grass, Fire, Water, Lightning, Psychic, Fighting, Darkness, or Metal Energy card.

Basic Pokémon Card: A card you can play directly from your hand on your turn. *See Evolution card.*

Bench: The place for your Pokémon that are in play but are not actively fighting. They come out and fight if the Active Pokémon retreats or is Knocked Out. When Benched Pokémon take damage, do not apply Weakness or Resistance.

Burn marker: What you put on a Pokémon to remind you it is Burned. Remove the marker if the Pokémon is Benched or Evolved. *See counter, damage counter.*

Counter: Object put on a Pokémon as a reminder (for example, a Char counter). A counter does not go away when you Bench the Pokémon, but it does go away if the Pokémon evolves (damage counters are a special exception to this rule). *See damage counter, Poison marker, Burn marker.*

Damage: What usually happens when one Pokémon attacks another. If a Pokémon has total damage greater than or equal to its Hit Points, it is Knocked Out.

Damage Counter: A counter put on your Pokémon to show it has taken damage. It stays on your Pokémon even if the Pokémon is Benched or evolved. Each damage counter counts as much damage as it has printed on it. *See counter, Poison marker.*

Defending Pokémon: Your Active Pokémon when your opponent is attacking.

Devolve: Certain cards can devolve an Evolved Pokémon, which is the opposite of evolving your Pokémon. When a Pokémon is devolved, it also loses Special Conditions and any other effects.

Discard Pile: The cards you have discarded. These cards are always face up. Anyone can look at these cards at any time.

Energy Card: Cards that power your Pokémon so they can attack. *See basic Energy card.*

Evolution Card: A card you play on top of a Basic Pokémon card (or on top of another Evolution card) to make it stronger.

Fossil Trainer cards: A special kind of Trainer card that acts like a Basic Pokémon when put into play. When a Fossil Trainer card is in your hand, deck, or discard pile, it is not considered a Basic Pokémon. However, these Trainer cards always count as Basic Pokémon during set-up.

Hit Points (HP): A number every Pokémon has, telling you how much damage it can take before it is Knocked Out.

In-Between Turns: The part of each turn when the game shifts from one player to the other. Check Poison, Burn, Asleep, and Paralysis at this step, and see whether any Pokémon are Knocked Out.

In Play: Your cards are in play when they are on the table. Basic Pokémon cards, Evolution cards, and Energy cards cannot be used unless they are in play. (The cards in your deck, your discard pile, and your Prizes are not in play, but your Benched Pokémon are.)

Knocked Out: A Pokémon is Knocked Out if it has damage greater than or equal to its Hit Points. That Pokémon goes to the discard pile along with all cards attached to it. When one of your opponent's Pokémon is Knocked Out, take one of your Prizes.

Item card: A type of Trainer card. Follow the instructions on the card and then discard it.

Lost Zone: Cards sent to the Lost Zone are no longer playable during that match. Put them face up anywhere out of play.

Owner: A Pokémon with a Trainer's name in its title, such as Brock's Sandshrew or Team Rocket's Meowth.

Poison Marker: Object put on a Pokémon to remind you it is Poisoned. Remove the marker if the Pokémon is Benched or evolved. *See counter, damage counter.*

Poké-Body: An effect that is active as soon as that Pokémon card is in play and lasts until the Pokémon leaves play.

Poké-Power: A once-per-turn power on Active and Benched Pokémon you must choose to use. Most Poké-Powers are turned off if the Pokémon has a Special Condition.

Pokémon: The colorful characters that fight for you in the Pokémon Trading Card Game. They are represented in the game by Basic Pokémon and Evolution cards.

Pokémon-ex: Pokémon-ex are a stronger form of Pokémon with a special drawback: when your Pokémon-ex is Knocked Out, your opponent draws two Prize cards instead of one.

Pokémon LEGEND: Special double cards that showcase powerful Legendary Pokémon. Both cards must be played together at the same time.

Pokémon LV.X: Stronger versions of a regular Pokémon, put on top of the regular Pokémon of the same name and adding extra abilities to the original Pokémon.

Pokémon Power: A special ability some Pokémon have. Pokémon Powers are divided into two categories: Poké-Power and Poké-Body. They always include the words "Poké-Power" or "Poké-Body" so you can tell they are not attacks.

Pokémon SP: A special Pokémon trained by a particular Trainer, with a symbol in its name to show its owner.

Pokémon Tool: A special kind of Trainer card (an Item) you can attach to your Pokémon to help you. Each Pokémon can have only 1 Pokémon Tool attached at any time.

Prize Cards: The 6 cards you put face down at the start of the game. Every time one of your opponent's Pokémon is Knocked Out, you take 1 of your Prizes into your hand (or 2 Prizes, for a Pokémon-ex). When you take your last Prize card, you win!

Resistance: A Pokémon with Resistance takes less damage when attacked by Pokémon of a certain type. The amount of Resistance is printed next to the type of Resistance(s) a Pokémon has, if any.

Retreat: When you switch your Active Pokémon with one of your Benched Pokémon. To retreat, you must discard Energy from the retreating Pokémon equal to the Retreat Cost of the Pokémon. This cost appears in the lower right-hand corner of the card. You can only retreat once per turn.

Special Conditions: Asleep, Burned, Confused, Paralyzed, and Poisoned are called Special Conditions.

Stadium Card: A type of Trainer card similar to an Item card, but stays in play after you play it. Only one Stadium card can be in play at a time—if a new one comes into play, discard the old one and end its effects. You can play only one Stadium card each turn.

Sudden Death: Sometimes both players win at the same time. In this case, you play a short game called "Sudden Death" (use only 1 Prize card each instead of 6).

Supporter Card: A Trainer card similar to an Item card. You can play only one Supporter card each turn.

Technical Machine: A kind of Trainer card (an Item) you can attach to your Pokémon. When attached, your Pokémon can use the Technical Machine attack as its own. Technical Machine cards remain attached unless the card text says otherwise.

Trainer Card: Special cards you play to gain advantages in the game. *See Item card, Stadium card, Supporter card.*

Trainers' Pokémon: Pokémon with Trainers' names in their titles, like Brock's Sandshrew. You cannot evolve a regular Sandshrew into Brock's Sandslash, and you cannot evolve a Brock's Sandshrew into a regular Sandslash. This is because "Brock's" is part of the name.

Weakness: A Pokémon with Weakness takes more damage when attacked by Pokémon of a certain type. The effect of the Weakness is indicated next to the type(s) of Weakness a Pokémon has, if any.

Appendix III: Team Details

Andrew Siddall served as the workflow leader for this portion of the project. Andrew understood the goal and did extensive research on the topic. Adlene, Chris, and Matt all contributed to this document as well. We all worked together by adding content, editing each other's work, and just bouncing ideas off each other. Because of this there was no one place that only one of us contributed. We all worked cooperatively and did our fair share of the work.

Appendix IV: Writing Center Report

Cal U Writing Center Report

Client: Adlene Bellaoucha
Staff or Resource: Autumn B.
Date: November 13, 2018, 5:00pm - 6:00pm

Did the student request that the instructor receive a visit report?: Yes

What course was serviced by this visit?: CSC 490

What goals were established for this tutoring session?: Revising;

How did the process of this consulting session address the established goals?: This was a twenty page senior project. It was not complete yet. I was only able to get to page thirteen. There were quite a few typos and inconsistencies.

Appendix V: Workflow Authentication

I, Andrew Siddall, hereby declare and confirm that I have performed the work as documented herein.

Signature: ____

Date: ____________________

I, Adlene Bellaoucha, hereby declare and confirm that I have performed the work as documented herein.

Signature: ____

Date: ____________________

I, Mathew Bedillion, hereby declare and confirm that I have performed the work as documented herein.

Signature: ____

Date: ____________________

I, Christopher Crisson, hereby declare and confirm that I have performed the work as documented herein.

Signature: [illegible]

Date: ______11/13/2018______________

Citations

General Python FAQ. (n.d.). Retrieved from https://docs.python.org/3/faq/general.html

What is open source? (n.d.). Retrieved from https://opensource.com/resources/what-open-source