Natalia Wilson
Professor Fanaei
EECE 2140
December 14, 2022

Elemental Heroes Final Project

For the final project, I created a text-based game called Elemental Heroes which was inspired by an existing card game called Card-Jitsu. The main features of my game are conditional statements, creating and manipulating arrays, functions, and incorporating the Random module. In addition, I imported new libraries to allow my game to be more user friendly.

The main concept of Elemental Heroes is that the user will be playing against the computer. The user and computer will receive four random cards that each consist of an element and a power rating. The elements can be fire, water, or snow, which can have a power rating ranging from one to ten. The user can pick a card from their deck to play against the computer. The winner of the battle is determined by whose element is more powerful. The winning combinations are: fire beats snow, water beats fire, and snow beats water. If the computer and user both play a card with the same element, the winner will be determined by who has the higher power rating. The game will continue until either the player or computer have run out of their three lives that were given to them at the beginning of the game. The user will receive a coin bonus worth two coins as a reward for completing the game. If the user won the game, the coin bonus begins at five coins and is multiplied by the number of lives the user had left from the previous game. These coins can be used in the shop to upgrade the power of their cards, increase the number of lives they begin with, or gamble to possibly win more coins.

In order to create Elemental Heroes, I had to plan out how to incorporate different aspects of what I learned throughout the semester into the game. I decided that arrays will be used to create and manipulate the decks for both the computer and user. To ensure that the decks between the user and the computer would be different, I used the Random module to create cards that have a random probability of choosing fire, water, or snow with a power rating between the range of 1 to 10.

Various conditional statements were used to advance the user throughout the game. The main type of conditional statements used are If statements to determine who won the round. The conditional statements checks to see if the elements are the same, and if that is true, a nested If statement would check whether the user has a higher power rating than the computer. If the elements are different, a set of If statements are used to check the winning combinations between the user's card and the computer's card. A crucial While loop was used to ensure that the game ends once the user’s lives or computer’s lives reach 0. Once the game ends, the user has the option to visit the shop, which is a function I created to add more elements to the game. The shop allows the user to use their coins to buy upgrades to make their deck stronger or gamble their coins by guessing a random number that the computer is not thinking of. All of the variables in the function are global so that it can be reflected back in the main body of the code.

To play my game, I used different libraries throughout the code that have to be installed before playing. I installed “from termcolor import colored” to add colors to the text to highlight important information and “import time” to slow down the text on the screen. Also, I used “import sys” but in most

systems that have Python installed, the module is already downloaded. Once the libraries are installed, simply run the Python file to begin playing.

When I first began to create the deck for the user, I had difficulties with how I wanted the deck to be formed. I originally had a four by two array with elements in the 0th column and the power rating in the 1st column. I wanted the user to choose their playing card and have it be displayed on the screen. The output of the card had extra characters, too many nested arrays, and it would not run the series of conditional statements to determine the winner correctly. I learned to convert the four by two array to a four by one array with both the element and power rating being one element rather than two elements by using the join() string method. If I had more time to spend on my project, I would create better visuals for my game. I would have liked to create a functioning GUI so that the user can click on the card that they want to play rather than entering a number. For future students in 2140, I would advise that they take time to truly learn the material that is taught because it can allow them to be more creative with their projects. I had an amazing time creating Elemental Heroes because I allowed my creativity to take over and I was not worried about if my idea was unattainable.

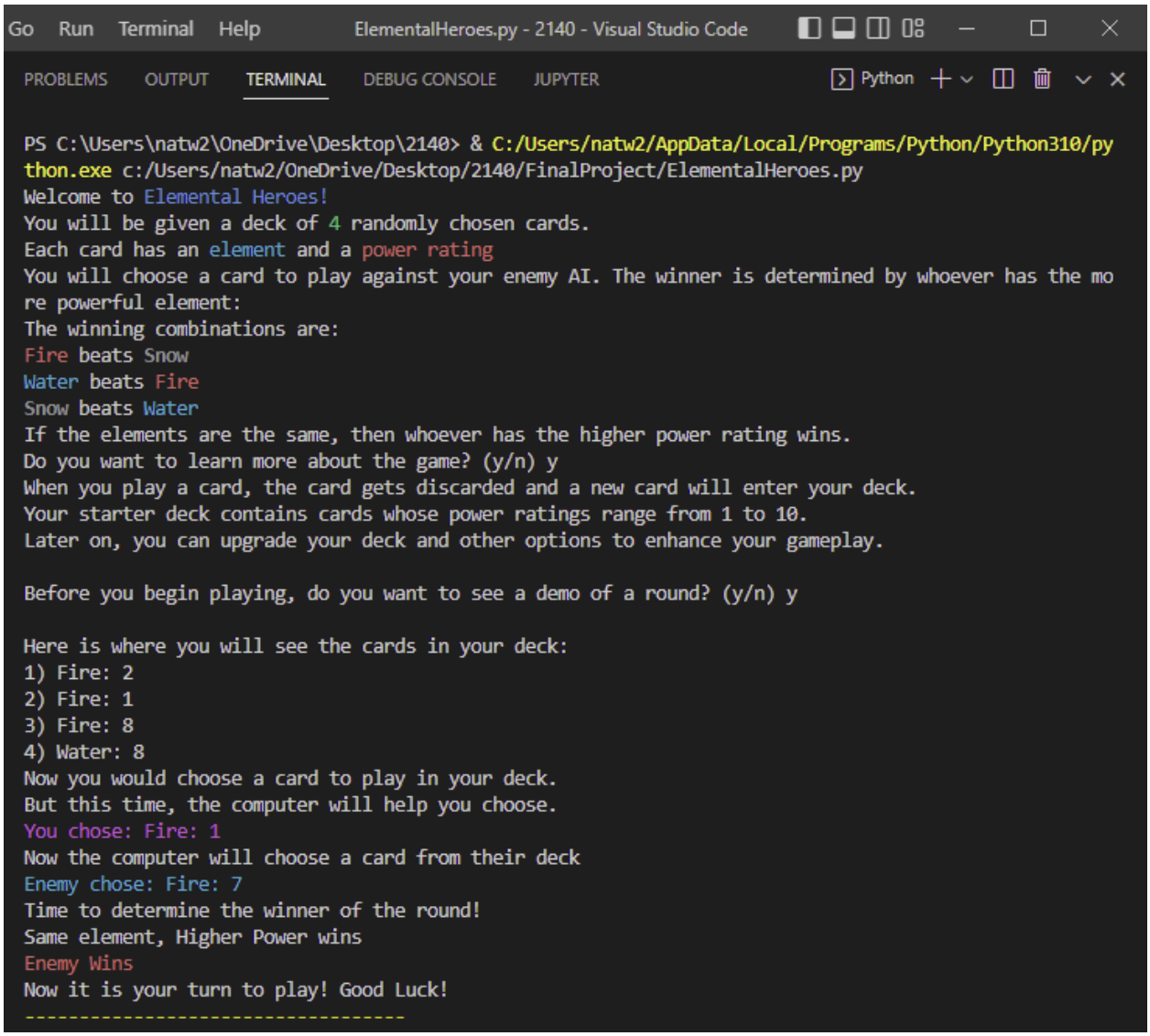
Appendix:
The Demo Code and the Results:

```
def demo():
    # CARD ELEMENTS
    elements = ["Fire", "Water", "Snow"]

    # USER STARTING DECK
    userDeck = []
    for i in range(4):
        curEle = elements[random.randint(0,2)]
        curPower = random.randrange(lowestPower, highestPower)
        indivCard = (str(curEle), str(curPower))
        card = ": ".join(indivCard)
        userDeck = np.append(userDeck, card)

    # ENEMY STARTING DECK
    enemyDeck = []
    for i in range(4):
        curEle = elements[random.randint(0,2)]
        curPower = random.randrange(lowestPower, highestPower)
        indivCard = (str(curEle), str(curPower))
        card = ": ".join(indivCard)
        enemyDeck = np.append(enemyDeck, card)

    # PRINTS USERS DECK
```

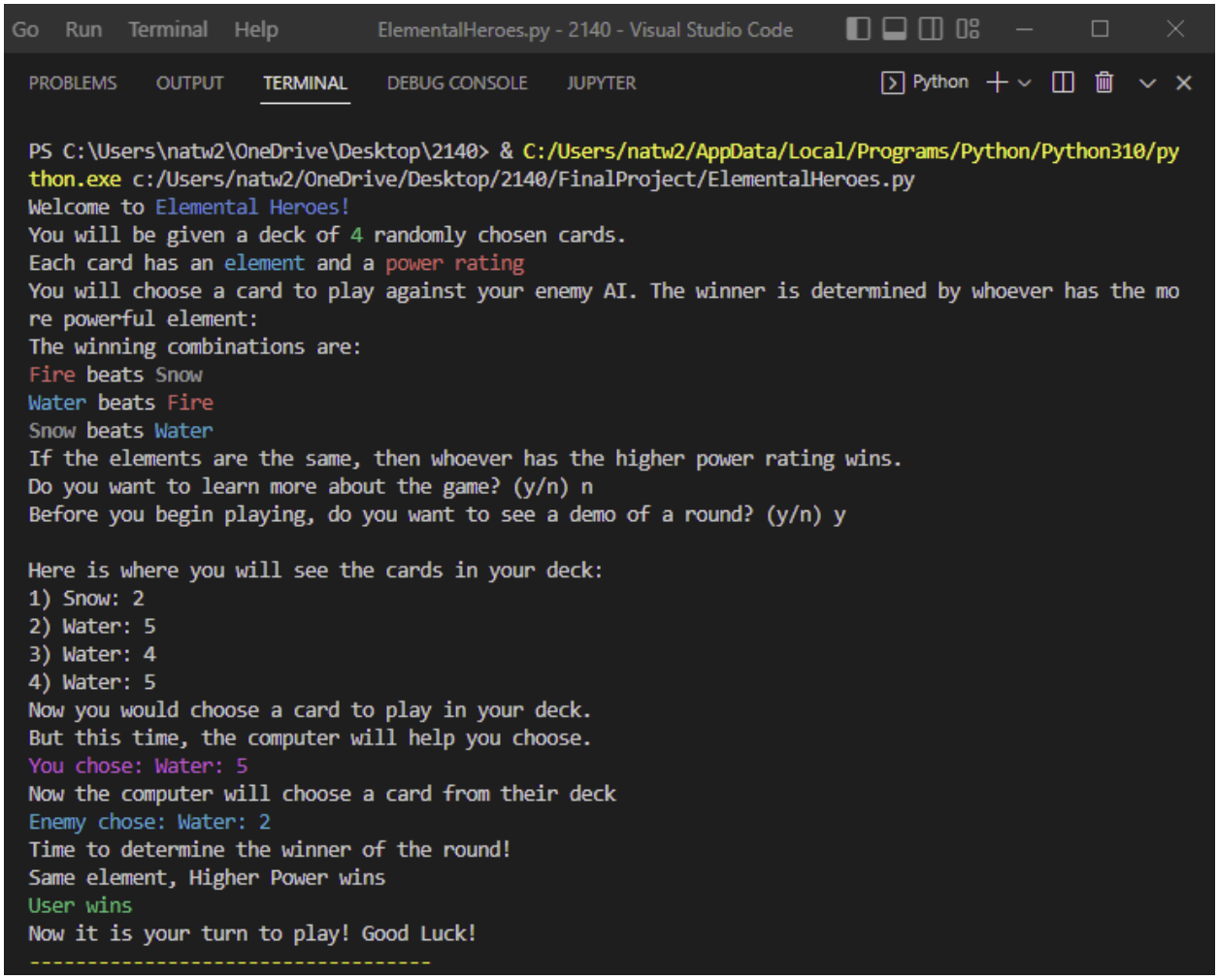
```
    print("\nHere is where you will see the cards in your deck: ")
    for i in range(4):
        print(f"{i+1}) {userDeck[i]}")
    time.sleep(3)

    # USER CHOOSES A CARD FROM THEIR DECK
    print("Now you would choose a card to play in your deck.\n\
But this time, the computer will help you choose.")
    time.sleep(4)
    chosenCard = random.randint(1,4)
    userCurCard = userDeck[chosenCard-1]
    print(colored(f"You chose: {userCurCard}", "magenta"))
    userCardSplit = userCurCard.split(':')
    time.sleep(1.5)

    # ENEMY CHOOSES CARD FROM THEIR DECK
    print("Now the computer will choose a card from their deck")
    enemyChosenCard = random.randint(0,3)
    enemyCurCard = enemyDeck[enemyChosenCard]
    elipses()
    print(colored(f"Enemy chose: {enemyCurCard}", "cyan"))
    sys.stdout.write('\x1b[2K')
    enemyCardSplit = enemyCurCard.split(':')
    time.sleep(1.0)

# DETERMINES WINNER OF THE ROUND
    print("Time to determine the winner of the round!")
    time.sleep(1.5)
    # IF ELEMENTS ARE THE SAME
    if userCardSplit[0] == enemyCardSplit[0]:
        print("Same element, Higher Power wins")
        time.sleep(0.85)
        if userCardSplit[1] > enemyCardSplit[1]:
            print(colored("User wins", "green"))
            time.sleep(1)
        elif userCardSplit[1] < enemyCardSplit[1]:
            print(colored("Enemy Wins", "red"))
            time.sleep(1)
        else:
```

```
            print("You have the same exact Card!! Its a draw this round.
")
            time.sleep(1)
    # FIRE BEATS SNOW
    elif userCardSplit[0] == "Fire" and enemyCardSplit[0] == "Snow":
        print("Fire beats Snow!")
        print(colored("User wins", "green"))
        time.sleep(1)
    elif userCardSplit[0] == "Snow" and enemyCardSplit[0] == "Fire":
        print("Fire beats Snow")
        print(colored("Enemy Wins", "red"))
        time.sleep(1)
    # WATER BEATS FIRE
    elif userCardSplit[0] == "Fire" and enemyCardSplit[0] == "Water":
        print("Water beats Fire!")
        print(colored("Enemy Wins", "red"))
        time.sleep(1)
    elif userCardSplit[0] == "Water" and enemyCardSplit[0] == "Fire":
        print("Water beats Fire!")
        print(colored("User wins", "green"))
        time.sleep(1)
    # SNOW BEATS WATER
    elif userCardSplit[0] == "Water" and enemyCardSplit[0] == "Snow":
        print("Snow beats Water!")
        print(colored("Enemy Wins", "red"))
        time.sleep(1)
    elif userCardSplit[0] == "Snow" and enemyCardSplit[0] == "Water":
        print("Snow beats Water!")
        print(colored("User wins", "green"))
        time.sleep(1)

    print("Now it is your turn to play! Good Luck!")
    time.sleep(1.25)

wantDemo = str(input("Before you begin playing, do you want to see a demo
of a round? (y/n) "))
if wantDemo.lower() == 'y':
    demo()
else:
    print("Good Luck!")
```

The demo code was run 2 times to show different results:
The first run and the results:



The second run and the results.



The Complete Python Code:

```
# Final Project Created by Natalia Wilson
# Inspiration: Card Jitsu, the Club Penguin Card Game

#Libraries
import numpy as np
import random
import sys
from termcolor import colored
import time

# SHOP COINS
coins = 0
coinsWon = 0

# LIVES
userLives = 3
enemyLives = 3
playAgain = 'y'
addUserLives = 0

# CARD POWER
lowestPower = 1
highestPower = 10

# ELIPSES FUNCTION
def elipses():
    print(colored("Enemy is choosing a card", "cyan"))
    for i in range(3):
        print("")
        sys.stdout.write('\x1b[1A')
        time.sleep(0.25)
        print(".")
        sys.stdout.write('\x1b[1A')
        time.sleep(0.25)
        print("..")
        sys.stdout.write('\x1b[1A')
        time.sleep(0.25)
        print("...")
        time.sleep(0.25)
```

```
        sys.stdout.write('\033[F')
        sys.stdout.write('\033[K')
    sys.stdout.write('\033[F')
    sys.stdout.write('\033[K')

#Introduction
text = colored("Elemental Heroes!", "blue")
print("Welcome to " + text)
text = colored("4", 'green')
print("You will be given a deck of " + text + " randomly chosen cards.")
text = colored("element", "cyan")
text2 = colored("power rating", "red")
print("Each card has an "+text+" and a " + text2 + "\n\
You will choose a card to play against your enemy AI. The winner is
determined by whoever has the more powerful element: \n\
The winning combinations are:")
textFire = colored("Fire","red")
textSnow = colored("Snow","grey", attrs=["bold", "dark"])
textWater = colored("Water", "cyan")
print(f"{textFire} beats {textSnow}\n{textWater} beats
{textFire}\n{textSnow} beats {textWater}")
print("If the elements are the same, then whoever has the higher power
rating wins. ")
# More information
moreInfo = str(input("Do you want to learn more about the game? (y/n) "))
if moreInfo.lower() == 'y':
    print("When you play a card, the card gets discarded and a new card
will enter your deck.\n\
Your starter deck contains cards whose power ratings range from 1 to
10.\n\
Later on, you can upgrade your deck and other options to enhance your
gameplay.\n")
# Demo
def demo():
    # CARD ELEMENTS
    elements = ["Fire", "Water", "Snow"]

    # USER STARTING DECK
    userDeck = []
    for i in range(4):
```

```
        curEle = elements[random.randint(0,2)]
        curPower = random.randrange(lowestPower, highestPower)
        indivCard = (str(curEle), str(curPower))
        card = ": ".join(indivCard)
        userDeck = np.append(userDeck, card)

    # ENEMY STARTING DECK
    enemyDeck = []
    for i in range(4):
        curEle = elements[random.randint(0,2)]
        curPower = random.randrange(lowestPower, highestPower)
        indivCard = (str(curEle), str(curPower))
        card = ": ".join(indivCard)
        enemyDeck = np.append(enemyDeck, card)

    # PRINTS USERS DECK
    print("\nHere is where you will see the cards in your deck: ")
    for i in range(4):
        print(f"{i+1}) {userDeck[i]}")
    time.sleep(3)

    # USER CHOOSES A CARD FROM THEIR DECK
    print("Now you would choose a card to play in your deck.\n\
But this time, the computer will help you choose.")
    time.sleep(4)
    chosenCard = random.randint(1,4)
    userCurCard = userDeck[chosenCard-1]
    print(colored(f"You chose: {userCurCard}", "magenta"))
    userCardSplit = userCurCard.split(':')
    time.sleep(1.5)

    # ENEMY CHOOSES CARD FROM THEIR DECK
    print("Now the computer will choose a card from their deck")
    enemyChosenCard = random.randint(0,3)
    enemyCurCard = enemyDeck[enemyChosenCard]
    elipses()
    print(colored(f"Enemy chose: {enemyCurCard}", "cyan"))
    sys.stdout.write('\x1b[2K')
    enemyCardSplit = enemyCurCard.split(':')
    time.sleep(1.0)
```

```
# DETERMINES WINNER OF THE ROUND
    print("Time to determine the winner of the round!")
    time.sleep(1.5)
    # IF ELEMENTS ARE THE SAME
    if userCardSplit[0] == enemyCardSplit[0]:
        print("Same element, Higher Power wins")
        time.sleep(0.85)
        if userCardSplit[1] > enemyCardSplit[1]:
            print(colored("User wins", "green"))
            time.sleep(1)
        elif userCardSplit[1] < enemyCardSplit[1]:
            print(colored("Enemy Wins", "red"))
            time.sleep(1)
        else:
            print("You have the same exact Card!! Its a draw this round.
")
            time.sleep(1)
    # FIRE BEATS SNOW
    elif userCardSplit[0] == "Fire" and enemyCardSplit[0] == "Snow":
        print("Fire beats Snow!")
        print(colored("User wins", "green"))
        time.sleep(1)
    elif userCardSplit[0] == "Snow" and enemyCardSplit[0] == "Fire":
        print("Fire beats Snow")
        print(colored("Enemy Wins", "red"))
        time.sleep(1)
    # WATER BEATS FIRE
    elif userCardSplit[0] == "Fire" and enemyCardSplit[0] == "Water":
        print("Water beats Fire!")
        print(colored("Enemy Wins", "red"))
        time.sleep(1)
    elif userCardSplit[0] == "Water" and enemyCardSplit[0] == "Fire":
        print("Water beats Fire!")
        print(colored("User wins", "green"))
        time.sleep(1)
    # SNOW BEATS WATER
    elif userCardSplit[0] == "Water" and enemyCardSplit[0] == "Snow":
        print("Snow beats Water!")
        print(colored("Enemy Wins", "red"))
```

```
        time.sleep(1)
    elif userCardSplit[0] == "Snow" and enemyCardSplit[0] == "Water":
        print("Snow beats Water!")
        print(colored("User wins", "green"))
        time.sleep(1)

    print("Now it is your turn to play! Good Luck!")
    time.sleep(1.25)

wantDemo = str(input("Before you begin playing, do you want to see a demo
of a round? (y/n) "))
if wantDemo.lower() == 'y':
    demo()
else:
    print("Good Luck!")

# SHOP FUNCTION, INCREASE LIVES, POWER RATINGS, OR GAMBLE COINS
def shop():
    global coins
    global userLives
    global highestPower
    global lowestPower
    global addUserLives
    coinsWon = 0
    print(colored("-----------------------------------", "yellow"))
    print(colored("Welcome to the Shop!", "blue"))
    text = colored(f"{coins}", "green")
    print("You currently have " + text +" coins")
    print("You can spend your coins to buy 2 more lives that costs 20
coins (1)\n\
You can spend your coins to buy more powerful cards that costs 10 coins
(2) OR\n\
You can gamble your coins for a chance to recieve more! (3)")
    shopChoice = int(input("What would you like to do? (1-3) "))
    while shopChoice > 3:
        print("Invalid Input: Choose a number between 1 and 3")
        shopChoice = int(input("What would you like to do? (1-3) "))
    if shopChoice == 1 and coins >= 20:
        addUserLives = 2
        coins -= 20
```

```
        print("You have successfully bought the power up!")
    elif shopChoice == 2 and coins >= 10:
        highestPower += 4
        lowestPower += 2
        coins -= 10
        print("You have successfully bought the power up!")
    elif shopChoice == 3:
        coinsWon = 0
        print("You have the choice to bet as many coins as you want with a
random multiplier.\n\
The catch is, if you guess the number that the dealer is thinking of, you
lose your coins!")
        numCoinGamble = int(input("How many coins are you willing to bet?
"))
        while numCoinGamble > coins:
            print("Invalid Amount: You cannot gamble more than what you
have.")
            numCoinGamble = int(input("How many coins are you willing to
bet? "))
        randmultiplier = random.randint(1,3)
        userNumGuess = int(input("Enter a number between 1 and 5 that you
think the dealer is not thinking of: "))
        while userNumGuess > 5 or userNumGuess < 0:
            print("Invalid Input")
            userNumGuess = int(input("Enter a number between 1 and 5 that
you think the dealer is not thinking of: "))
        dealerNum = random.randint(1,5)
        if userNumGuess == dealerNum:
            text = colored(f"{numCoinGamble}", "red")
            print(f"Uh Oh, You have guessed the dealer's number... you
have lost " + text + " coins")
            coins -= numCoinGamble
        else:
            print(f"You Won! The dealers number was {dealerNum}.")
            coinsWon += (numCoinGamble * randmultiplier)
            print(f"You have won {coinsWon} coins!")
    else:
        print("You dont have enough coins to purchase this.")
        print("Come back next time!")
    coins += coinsWon
```

```
    print(colored("Thank you for supporting the shop!", "cyan"))


# CARD ELEMENTS
elements = ["Fire", "Water", "Snow"]

# USER STARTING DECK
userDeck = []
for i in range(4):
    curEle = elements[random.randint(0,2)]
    curPower = random.randrange(lowestPower, highestPower)
    indivCard = (str(curEle), str(curPower))
    card = ": ".join(indivCard)
    userDeck = np.append(userDeck, card)

# ENEMY STARTING DECK
enemyDeck = []
for i in range(4):
    curEle = elements[random.randint(0,2)]
    curPower = random.randrange(lowestPower, highestPower)
    indivCard = (str(curEle), str(curPower))
    card = ": ".join(indivCard)
    enemyDeck = np.append(enemyDeck, card)

print(colored("-----------------------------------", "yellow"))

# GAME BEGINS
while playAgain.lower() == 'y':
    enemyLives = 3
    userLives = 3 + addUserLives
    while userLives != 0 and enemyLives != 0:

        # PRINTS LIVE COUNTS FOR USER
        text = colored(f"{userLives}", "green")
        print(f"User Lives Left: {userLives}")
        text = colored(f"{enemyLives}", "red")
        print(f"Enemy Lives Left: {enemyLives}")

        # PRINTS USERS DECK
        for i in range(4):
```

```
            print(f"{i+1}) {userDeck[i]}")

        # USER CHOOSES A CARD FROM THEIR DECK
        chosenCard = input("Choose a card from your deck (1-4): ")
        userInputInvalid = True
        while userInputInvalid:
            if str(chosenCard).isalpha():
                print("You entered a letter when asked to enter a
number.")
                chosenCard = input("Please Choose a card between 1 and 4:
")
            elif int(chosenCard) > 4:
                print("You chose a card outside the range of 1-4")
                chosenCard = input("Please Choose a card between 1 and 4:
")
            elif int(chosenCard) < 5:
                chosenCard = int(chosenCard)
                userInputInvalid = False

        userCurCard = userDeck[chosenCard-1]
        print(colored(f"You chose: {userCurCard}", "magenta"))
        userCardSplit = userCurCard.split(':')
        time.sleep(0.5)

        # ENEMY CHOOSES CARD FROM THEIR DECK
        enemyChosenCard = random.randint(0,3)
        enemyCurCard = enemyDeck[enemyChosenCard]
        elipses()
        print(colored(f"Enemy chose: {enemyCurCard}", "cyan"))
        sys.stdout.write('\x1b[2K')
        enemyCardSplit = enemyCurCard.split(':')
        time.sleep(1.5)

    # DETERMINES WINNER OF THE ROUND
        # IF ELEMENTS ARE THE SAME
        if userCardSplit[0] == enemyCardSplit[0]:
            print("Same element, Higher Power wins")
            time.sleep(0.85)
            if userCardSplit[1] > enemyCardSplit[1]:
                print(colored("User wins", "green"))
```

```
                time.sleep(1)
                enemyLives = enemyLives - 1
            elif userCardSplit[1] < enemyCardSplit[1]:
                print(colored("Enemy Wins", "red"))
                time.sleep(1)
                userLives = userLives - 1
            else:
                print("You have the same exact Card!! Its a draw this
round. ")
                time.sleep(1)
        # FIRE BEATS SNOW
        elif userCardSplit[0] == "Fire" and enemyCardSplit[0] == "Snow":
            print("Fire beats Snow!")
            print(colored("User wins", "green"))
            time.sleep(1)
            enemyLives = enemyLives - 1
        elif userCardSplit[0] == "Snow" and enemyCardSplit[0] == "Fire":
            print("Fire beats Snow")
            print(colored("Enemy Wins", "red"))
            time.sleep(1)
            userLives = userLives - 1
        # WATER BEATS FIRE
        elif userCardSplit[0] == "Fire" and enemyCardSplit[0] == "Water":
            print("Water beats Fire!")
            print(colored("Enemy Wins", "red"))
            time.sleep(1)
            userLives = userLives - 1
        elif userCardSplit[0] == "Water" and enemyCardSplit[0] == "Fire":
            print("Water beats Fire!")
            print(colored("User wins", "green"))
            time.sleep(1)
            enemyLives = enemyLives - 1
        # SNOW BEATS WATER
        elif userCardSplit[0] == "Water" and enemyCardSplit[0] == "Snow":
            print("Snow beats Water!")
            print(colored("Enemy Wins", "red"))
            time.sleep(1)
            userLives = userLives - 1
        elif userCardSplit[0] == "Snow" and enemyCardSplit[0] == "Water":
            print("Snow beats Water!")
```

```
            print(colored("User wins", "green"))
            time.sleep(1)
            enemyLives = enemyLives - 1

        print(colored("-----------------------------------", "yellow"))

        # UPDATES DECK FOR USER AND ENEMY
        userDeck = np.delete(userDeck, chosenCard-1)
        curEle = elements[random.randint(0,2)]
        curPower = random.randrange(lowestPower, highestPower)
        indivCard = (str(curEle), str(curPower))
        card = ": ".join(indivCard)
        userDeck = np.append(userDeck, card)

        enemyDeck = np.delete(enemyDeck, enemyChosenCard)
        curEle = elements[random.randint(0,2)]
        curPower = random.randrange(lowestPower, highestPower)
        indivCard = (str(curEle), str(curPower))
        card = ": ".join(indivCard)
        enemyDeck = np.append(enemyDeck, card)

    # CHECKS LIFE COUNT, RECIEVE COINS, AND ENDS GAME
    if userLives == 0:
        print("Oh No! You Lost to the enemy!!!")
        text = colored("2", "green")
        print("You earned " + text + " coins for completing the game.")
        coins += 2
    elif enemyLives == 0:
        print("Congratulations You Won the Game!!!")
        coinsWon += (5 * userLives)
        coins += coinsWon
        text = colored(f"{coinsWon}", "green")
        print(f"You earned " + text + " coins for winning the game!")
        text = colored(f"{coins}", "green")
        print("You now have "+ text + " coins!")

    # SHOP TO USE COINS
    goToShop = str(input("Do you want to visit the shop to spend your
coins? (y/n) "))
    if goToShop.lower() == 'y':
```

```
        shop()

    # ASK USER IF THEY WANT TO PLAY AGAIN
    playAgain = str(input("Do you want to play again? (y/n) "))

# CREDITS
print(colored("Thanks for playing!", "cyan"))
```