



Wild Card Bot

Sprint 3 Retrospective

Team 25 - Wild Card Bot

Konstantin Liehr, Arun Thirumavalavan, Nikolas Damalas,
Kyle Arrowood, Michael Gu, Matthew Wang

What went well?

User Story #1

As a user, I want to see a Pokemon's basic information given its Name or Number

#	Description	Estimated Time	Owner
1	Figure out how to connect to Pokemon API	1 Hour	Nikolas
2	Create functions to parse the Pokemon's type, ability, and statistics	1.5 Hour	Nikolas
3	Create function to get the Pokemon's sprite	45 minutes	Nikolas
4	Create function to pass the pokemons name or number to the API call from the user to get the desired information	1 Hour	Nikolas
5	Create manual tests	45 minutes	Nikolas

Completed:

Using the !poke command along with a pokemon name or number works. It will display the pokemon, its base statistics, type(s), and abilities.

User Story #2

As a user, I want to see a shiny Pokemon's sprite

#	Description	Estimated Time	Owner
1	Create a function to get the shiny sprite of a Pokemon given its name or number	30 minutes	Nikolas
2	Create a function to allow users to compare a pokemon's normal and shiny sprites side by side	45 minutes	Nikolas
3	Create manual tests	45 minutes	Nikolas

Completed:

Using the !shiny command plus a pokemon name or number will display the “shiny” (rare alternate coloring) of a pokemon. Using “!shiny compare” plus a valid name or number will present both the normal and shiny sprites so the user can see the difference. These had to be put on two separate cards since 2 images cannot be put on the same embed in discord.

User Story #3

As a user, I want to see information about a Pokemon Item

#	Description	Estimated Time	Owner
1	Create a function to get the requested items information and sprite from the Pokemon API	1 Hour	Nikolas
2	Create function to list which Pokemon hold the item naturally, if any at all	30 minutes	Nikolas
3	Create manual test	1 Hour	Nikolas

Completed:

Using the !item command with an item name will display a card with the image of the item and its effect. Using “!item held” with an item name will list any Pokemon that can be found holding the item in the wild if possible.

User Story #4

As a user, I want to see information about Pokemon Moves

#	Description	Estimated Time	Owner
1	Create a function to get the requested moves information (name, type, power, added effects, flavor text, priority, effect chance, and accuracy) from the API	1.5 Hour	Nikolas
2	Create a function to list all the moves a specific Pokemon can know given its name or number.	2 Hour	Nikolas
2	Create manual tests	1 Hour	Nikolas

Completed:

Using the !move command with a Pokemon move name will list all relevant info about the move, such as type, power, accuracy, priority, and any added effects. If there are no effects then the output will say that. Using “!move” with a Pokemon name will list all the moves that Pokemon can learn in alphabetical order.

User Story #5

As a user, I want to see information about a Pokemon Ability

#	Description	Estimated Time	Owner
1	Create a function to get the requested abilities information (name and effect) from the API.	45 minutes	Nikolas
2	Create optional choice to list all Pokemon that can have the given ability	1 Hour	Nikolas
3	Create manual tests	45 minutes	Nikolas

Completed:

Using !ability plus an ability name will give a card describing what the ability does. “!ability all” plus an ability name will list all Pokemon that can have that ability.

User Story #6

As a user, I want to see evolutionary family of a Pokemon

#	Description	Estimated Time	Owner
1	Create a function to get the evolutionary family of a pokemon (what it evolves from and into)	2 Hour	Nikolas
2	Determine how to display the evolutionary family in the Discord chat.	1 Hour	Nikolas
3	Determine and List the evolution methods next to each stage in the evolutionary family.	2 Hour	Nikolas

4	Create manual tests	1 Hour	Nikolas
---	---------------------	--------	---------

Completed:

Using the “!evo” command plus a Pokemon name will display a text tree showing the evolutionary family of that Pokemon, plus an image of the Pokemon. The tree displays all the necessary requirements for each step of the evolutionary family.

User Story #7

As a user, I want to know what types are effective against other types.

#	Description	Estimated Time	Owner
1	Get a given types offensive capabilities (which types they do no damage, reduced damage, normal damage, and super effective damage to)	45 minutes	Nikolas
2	Get a given types defensive capabilities (which types do no damage, reduced damage, normal damage, and super effective damage to it)	45 minutes	Nikolas
3	Create a function to show the damage relations between only two types.	45 minutes	Nikolas
4	Create manual tests	1 Hour	Nikolas

Completed:

Using “!type” plus a Pokemon type will display all of the unique type interactions that type has both offensively and defensively with other types. Using “!type” plus two types will display the type matchup between those two types.

User Story #8

As a user, I want to know what a Pokemon’s egg group is and what it can breed with.

#	Description	Estimated Time	Owner
---	-------------	----------------	-------

1	Get a given Pokemon's egg group(s)	30 minutes	Nikolas
2	Get all other Pokemon in the egg group(s)	30 minutes	Nikolas
3	Create a function to display the egg group(s) and compatible Pokemon.	1 Hour	Nikolas
4	Create manual tests	30 minutes	Nikolas

Completed:

Using the "!egg" command with a Pokemon will display all the egg groups that the Pokemon is compatible with. Using "!egg" with an egg group name will list all the Pokemon found in that egg group. We could not display all the compatible Pokemon when displaying a Pokemon's egg groups because of limits on the amount of text that can be in a discord embed.

User Story #9

As a user, I want to know what a Pokemon's nature does

#	Description	Estimated Time	Owner
1	Get the stat boosts and debuffs of a given nature	30 minutes	Nikolas
2	Create a function to list all natures with a desired bonus, drawback, or both.	1.5 Hour	Nikolas
3	Create manual tests	1 Hour	Nikolas

Completed:

Using the "!nature" command with a Pokemon nature will display the information about that nature. If you use "!nature" and either a + or - with a Pokemon stat, it will list all natures that affect the desired stat in that manner. Using "!nature" with both a + in one stat and a - for another will list the stat that fulfills both of those stat changes if it exists.

User Story #10

As a user, I want to be able to view live sports scores on the discord server.

#	Description	Estimated Time	Owner
1	Create a function to search and find the requested sports game/event.	1 Hour	Kyle
2	Create a function to retrieve the current score.	1 Hour	Kyle
3	Create a function to retrieve the current time remaining.	1 Hour	Kyle
4	Create manual tests	30 Minutes	Kyle

Completed:

I completed this user story by using the command named “!score” the purpose of the “!score” command is to easily search an upcoming game or a live game of a specific team. When using the command if you pass it the argument of a team name that is currently in a game it will tell you the live score and time remaining in the event.

User Story #11

As a user, I want to view betting odds, spreads, and any other relevant information related to the sports betting for that event.

#	Description	Estimated Time	Owner
1	Create a function to retrieve the website with the betting information.	30 Minutes	Kyle
2	Create a function to scrape the different information such as betting odds, spread, over/under, etc.	3 Hours	Kyle
3	Create a function to display the info to the user.	30 Minutes	Kyle
4	Create manual tests	30 Minutes	Kyle

Completed:

This user story also uses the same command as user story 10 which is the “!score” command. It is also used in the same way. If you give it an argument of a team that is currently not playing a game live and has an upcoming game, it will display the time and date of when the event will

occur as well as all of the betting odds and information that you would need to know about the event.

User Story #12

As a user, I want to view players' season averages and career averages.

#	Description	Estimated Time	Owner
1	Create a function to retrieve the website with the player information.	30 Minutes	Kyle
2	Create a function to scrape the different statistics for basketball players.	3 Hours	Kyle
3	Create a function to scrape different statistics for football players.	3 Hours	Kyle
4	Create a function to display the info to the user.	30 minutes	Kyle
5	Create manual tests	30 Minutes	Kyle

Completed:

For this user story I created a command called “!player” that when given a valid player on ESPN’s database will return their current season averages assuming that they are a current player, or their most recent career averages assuming that they are a retired player. It will also give career averages and stats for the player and give a link to their ESPN webpage.

User Story #13

As a user, I want to get answers for simple math equations such as addition, subtraction, multiplication, division, and exponents.

#	Description	Estimated Time	Owner
1	Create a function to parse through the math equation.	3 Hours	Kyle
2	Create a function to solve the math equation.	2 Hours	Kyle
3	Create a function to display the	30 Minutes	Kyle

	results.		
4	Create manual tests	30 Minutes	Kyle

Completed:

For this user story I created a command named “!calc”. This command will take any math equation and give the solution to it. It also can support algebraic equations that solve for a variable.

User Story #14

As a user, I want to get answers for trigonometric equations quickly. These could include functions such as sine, cosine, and tangent.

#	Description	Estimated Time	Owner
1	Create a function to parse through the math equation, but now with trigonometric principles.	1 Hour	Kyle
2	Create a function to solve the math equation.	2 Hours	Kyle
3	Create a function to display the results.	30 Minutes	Kyle
4	Create manual tests	30 Minutes	Kyle

Completed:

This user story uses the same command as user story 13, the “!calc” command. The “!calc” command has the ability to calculate any trigonometric principles such as anything involving sine, cosine, tangent, etc.

User Story #15

As a user, I want to solve calculus problems quickly, these could include both derivatives and integrals.

#	Description	Estimated Time	Owner
1	Create an option for users to pick	30 Minutes	Kyle

	either a derivative or an integral.		
2	Create a function to handle the derivative including all of the options the user needs to give the bot in order to solve the problem.	30 Minutes	Kyle
3	Create a function to handle the integral including all of the options the user needs to give the bot in order to solve the problem.	30 Minutes	Kyle
4	Create a function to solve a derivative.	2 Hours	Kyle
5	Create a function to solve an integral.	2 Hours	Kyle
6	Create a function to display the results.	30 Minutes	Kyle
7	Create manual tests	30 Minutes	Kyle

Completed:

For this user story, I created two different commands, one for derivatives and one for integrals. First, the command to solve a derivative with respect to x is "!derive". The derive command will take an equation and return the derivate of it with respect to x. The command to solve integrals is "!integrate", this command supports but indefinite and definite integrals. To solve an indefinite integral with respect to x, the user will input the equation directly after the command as arguments. To solve a definite integral with respect to x, the user must give the arguments in the form: "!integrate definite <a> <equation>" where a and b are the lower abd upper bounds of the definite integral.

User Story #16

As a user, I want to start a blackjack game if no one is playing or redisplay the game if someone is currently playing.

#	Description	Estimated Time	Owner
1	Research the rules of blackjack and decide on the most efficient way to implement blackjack.	1 Hour	Matthew
2	Create a Card class that represents a	1 Hour	Matthew

	card. Include the possible suits and numbers as well as a text representation of the card.		
3	Create a Game class that represents a game and includes all fields necessary to monitor the game.	1 Hour	Matthew
4	Create a Player class that represents a player and keeps track of the player's currency, hand value, and other required fields.	1 Hour	Matthew
5	Create a function that checks for an existing Game object in the server and if it exists delete the old message representing the game and display a new message with the game info.	1 Hour	Matthew
6	Create a function that creates a Game object and creates a Player object for the caller of the command.	1 Hour	Matthew
7	Create a function that takes a Game object and creates a message displaying the game state.	2 Hours	Matthew
8	Create manual test cases.	1 Hour	Matthew

Completed:

For this user story, I initially created three classes: the Card class, the Player class, and the Game class. The Card class, which represents a playing card, consists of a suit and a value field. The Player class, which represents a player, consists of a field to keep track of the player's balance, a field representing the player's hand which is a list of Card objects, a field for the player's hand value, a field for the player's bet amount, and other additional overhead. It also includes methods to modify the player's balance and calculate the player's hand value. The Game class, which represents a blackjack game, consists of a field representing the players which is a list of Player objects, fields for the dealer's hand and value, a field for the game message, and other additional overhead. It includes methods to modify the game message, run the dealer's turn, and initialize the round. After setting up the required classes, I created two dictionaries to keep track of all the players and games, where each user has a Player object and each server has a Game object. I then created the '!blackjack' command, which allows users to start a new blackjack game on their server and add himself or herself as a player or redisplay the blackjack game if it already exists on the server. I first check to see if the server the command came from has a Game object. If it does, the game message is obtained from the

Game object, deleted from the text channel, and then another message with the game message's embed is sent to the text channel the command came from and assigned to be the new game message. If it does not, a Game object is created for the server and a Player object is created for the user who called the command, and the Player object is added to the player list of the Game object.

User Story #17

As a user, I want to join an existing blackjack game in session with a reaction.

#	Description	Estimated Time	Owner
1	Create a function that checks the reactions of the message representing the game between rounds.	1 hour	Matthew
2	Create a function that determines which users have reacted but are not in the game. It will create a player object for such users and add them to the list of players in the game.	1 hour	Matthew
3	Create manual test cases.	30 Mins	Matthew

Completed:

For this user story, I created a way for users to join a blackjack game by reacting with the black joker reaction. Upon creating or redisplaying the game, the bot will react to the game message with the black joker reaction and indicate it as the reaction to join the game. At the start of every round, a function will be called that obtains the Message object of the game message, extracts all users who have reacted to the black joker reaction from the Message object, then add all users that are currently not playing the blackjack game.

User Story #18

As a user, I want the bot to continuously host rounds of blackjack while the game is active and display the results of each round.

#	Description	Estimated Time	Owner
1	Create a function that starts a round by deciding the cards of each player and the dealer.	1 Hour	Matthew

2	Create a function that checks to see if the number of players is above zero and continuously calls the function that starts a round when the previous one ends.	1 Hour	Matthew
3	Create a function that takes a Game object and displays the results of the current round.	2 Hours	Matthew
4	Create manual test cases.	30 Mins	Matthew

Completed:

For this user story, I created a function that edits the game message to display the round results. It obtains the Message object of the game message, parses the result of each player in the Game object's list of players, and then constructs the embed that reflects the result of each player. The message is then edited with the new embed constructed, displaying the round results on the game message. I also implemented the while loop that runs the rounds continuously. Inside of the while loop, there are calls to the different stages of the round, including round initialization, bet change, running the round, and finally displaying the round results.

User Story #19

As a user, I want to hit, stand, or fold with reactions via a direct message.

#	Description	Estimated Time	Owner
1	Create a function that sends a direct message to the user with their current hand and the dealer's cards.	1 Hour	Matthew
2	Decide on the reactions to be used for hit, stand, and fold. Make sure the bot reacts to the direct message with these reactions when appropriate.	30 Mins	Matthew
3	Create a function that handles a hit and adds another randomly selected card to the user's hand. It should also check for busts and keep the value of ace in check.	1 Hour	Matthew
4	Create a function that handles a	1 Hour	Matthew

	stand. It should take the value of the player's hand and notify the Game object that the player is done with the round.		
5	Create a function that handles a fold. It should notify the Game object the player did not play the round.	1 Hour	Matthew
6	Create manual test cases.	30 Mins	Matthew

Completed:

For this user story, I created four functions: one that sends a direct message with the player's hand obtained from the Player object, one that handles the hit functionality and direct messages the user with his or her new hand, one that handles the stand functionality, and one that handles the stand functionality. Initially, the function that sends the player's hand in a direct message is executed. Within the function, I also make the bot react with the green checkmark, green cross, and cancel reactions which correspond to hit, stand, and fold accordingly. If the user hits, then the hit function is called, which adds a randomly generated card to the user's hand stored in the Player object, checks if the user's hand exceeds 21 and busts, and sends another direct message with the player's new hand if the player does not bust. The bot will then react with the green checkmark and green cross for hit and stand. If the player busts, a direct message indicating the user busted is sent and the user is notified he or she lost the round. If the user stands, then the stand function is called, which finalizes the user's hand for the round and sends a direct message indicating the user's decision to stand. If the user folds, then the fold function is called, which sets the hand value of the Player object corresponding to the user to -1 and sends a direct message indicating the user's decision to fold. A hand value of -1 is an indication to the function that compares the hands of the players with the dealer that the player should not be considered for the round.

User Story #20

As a user, I want to make or change my bet at specific amounts with reactions at the beginning of each round.

#	Description	Estimated Time	Owner
1	Decide on the bet amounts and reactions to be used for the different amounts. Make sure the bot reacts to the direct message with these reactions when appropriate.	30 Mins	Matthew

2	Create a function that sends a timed direct message before every round indicating if the user would like to change their bet.	1 Hour	Matthew
3	Create manual test cases.	30 Mins	Matthew

Completed:

For this user story, I implemented a function that constructs the direct message that allows the user to change his or her bet and sends it to the user. This includes making the bot react with the keycap one, two, five, and ten reactions and the display of the bet amounts associated with each reaction. It then takes the reaction the user reacted to the direct message with and adjusts the bet amount accordingly. This is called by the while loop that runs the rounds after round initialization.

User Story #21

As a user, I want the bot to keep track of the currency.

#	Description	Estimated Time	Owner
1	Create a function that compares the hand value of the player to the dealer and determines if the player won or not.	1 Hour	Matthew
2	Create a function that adjusts the currency balance of a player. It considers whether the player won or not and the bet amount.	1 Hour	Matthew
3	Create a function that directly messages the current currency balance of a player to the player.	30 Mins	Matthew
4	Create a function that detects if the player's currency falls below zero and directly messages the user that they have bankrupted. Resets the player's currency to the default amount.	30 Mins	Matthew
5	Create manual test cases.	30 Mins	Matthew

Completed:

For this user story, I added the 'balance' argument to the '!blackjack' command that will directly message the user his or her balance upon calling the command. The balance is stored in the Player object. I also implemented a function that checks the final hand value of all players against the dealer's final hand value and adjusts the balance according to whether they won, lost, or tied. The function also sends a direct message to each player, which displays his or her round results. I also added to the method in the Player class that adjusts the player's balance a if loop to detect if the balance of the player falls below zero. If it does, then a direct message is sent to the user and the currency is reset to \$100,000, the default amount.

User Story #22

As a user, I want to be able to leave the game in between rounds.

#	Description	Estimated Time	Owner
1	Enhance the function that sends a timed direct message before every round to change bet amount to include a reaction to leave the game.	30 Mins	Matthew
2	Create a function that removes the Player object from the Game object's list of players, removing the player from the game.	1 Hour	Matthew
3	Create manual test cases.	30 Mins	Matthew

Completed:

For this user story, I modified the function that directly messages the user the ability to change his or her bet. On top of the four previous reactions, I made the bot react with the red cross reaction. I then added another if loop to detect if the reaction received from the user is the red cross reaction. If the user does react with the red cross reaction, the Player object of the player is removed from the Game object, indicating that the player has left the game. A direct message will also be sent to the player indicating that they are no longer part of the game.

User Story #23

As a user, I want to know the exact rules being used for the blackjack game.

#	Description	Estimated Time	Owner
1	Create a function that directly	1 Hour	Matthew

	messages the rules to the user. The rules should be organized into sections and the user can request just one section.		
5	Create manual test cases.	30 Mins	Matthew

Completed:

For this user story, I added the 'rules' argument to the '!blackjack' command that will directly message the user the five sections of the rules upon calling the command. I then added the ability to specify an additional argument that will only send one section of the rules given that the argument matches a section name. If the section name given via the argument does not exist, the user is sent a direct message indicating the section does not exist and a list of all the section names.

User Story #24

As a user, I want to display information and stats from my Steam account.

#	Description	Estimated Time	Owner
1	Research how the Steam API works and determine how to extract profile and game stats	2 Hr	Michael
2	Create a function that displays status of Steam account	1 Hr	Michael
3	Create a function that displays the game a user is currently playing	1 Hr	Michael
4	Create a function that displays the location of a user	1 Hr	Michael
5	Create a function that displays the groups a user is apart of	1 Hr	Michael
6	Create a function that displays a profile card of a user	3 Hr	Michael
7	Create manual test cases	1 Hr	Michael

Completed:

The Steam profile functions display various information on a specified steam user. The “!status” function displays the current status (online/offline) status of a user. The “!location” function displays the current location of a user which is taken from their profiles. The “!profile” function outputs a profile card of a user containing their profile picture, status, steamID, location, currently playing game, and steam groups they are a part of. All data was pulled from the steam profile API.

User Story #25

As a user, I want to display information and stats of a game on Steam.

#	Description	Estimated Time	Owner
1	Create a function that displays a description of a game	2 Hr	Arun
2	Create a function that displays the price of a game	2 Hr	Arun
3	Create a function that displays the publishers and developers of a game	2 Hr	Arun
4	Create a function that displays a profile card of a game	3 Hr	Arun
5	Create manual test cases	1 Hr	Arun

Completed:

The steam game profile function displays various stats including the picture of the splashart, if the game is free, the name, its app id etc. when the command !steam game <game name> is called.

User Story #26

As a user, I want to display top charts of games on steam.

#	Description	Estimated Time	Owner
1	Create a function that links multiple databases containing Steam game stats.	1 Hr	Konstantin
2	Create a function that displays a list of trending games on steam	3 Hrs	Konstantin

3	Create a function that displays a list of recently added games on steam	3 Hrs	Konstantin
4	Create a function that displays a list of top played games on steam	3 Hrs	Konstantin
5	Create manual test cases	1 Hr	Konstantin

Completed:

All of the steam commands implemented in #26 use information from SteamSpy which is a database website that collects information about the game sales and play on Steam. The command “!steam trending” displayed the top 10 trending games in the last 2 weeks on Steam and “!steam recent” displays the top 10 recently added games that are also somewhat popular on release in measure of playtime by users. The “!steam top” command displays the top 10 most played games in the last 2 weeks using the steamspy API to do so.

User Story #27

As a user, I want to be able to choose my class at the beginning of the game.

#	Description	Estimated Time	Owner
1	Create a function that starts the RPG game and notifies the user after starting.	1 Hr	Michael
2	Create a function that prompts the user to choose between classes.	1 Hr	Michael
3	Create a function that stores the data of the chosen class for the character specific to that user.	2 Hr	Michael
4	Create manual test cases	1 Hr	Michael

Completed:

The “!rpg start” function starts the rpg. The bot will notify the user that the game has started and output a start menu containing info on each class the user can choose. There will be emojis the user can react with to choose their class. Upon choosing a class, the bot will output the class the user has chosen and prompt them for a character name. Upon entering a name, the bot will output the name the user has chosen and also display the help menu.

User Story #28

As a user, I want to be able to check my status which should include a short summary of my character's stats and their items currently.

#	Description	Estimated Time	Owner
1	Create a text file and dictionary which holds profiles of players. Create a function that reads the text file into the dictionary	1 Hr	Michael
2	Create a function that displays the profile of a player	1 Hr	Michael
3	Create a function that displays the stats of a character	2 Hr	Michael
4	Create manual test cases	1 Hr	Michael

Completed:

The "!rpg myinfo" function will display information containing the name, discord ID, class, and balance of a user's character. The "!rpg stats" function will display information containing the stats of a user's character. This includes hp, armor, attack damage, and other stats. A help function, "!rpg help" was also created for every rpg command which includes their usage.

User Story #29

As a user, I want to be able to check my inventory and switch items or discard them.

#	Description	Estimated Time	Owner
1	Create a section of the text file that stores inventory.	30 min	Arun
2	Create a function that shows the user their inventory.	2 Hr	Michael
3	Create a function that discards a certain item if the user specifies it and notifies the user.	2 Hr	Arun
4	Create manual test cases	1 Hr	Arun

Completed:

When the user types in the command !rpg myequipment the bot will send back a message with the currently equipped items in both slots on the player's character. The "!rpg inventory" function displays the inventory of the user which contains their items they have bought. The !rpg sell" function handles selling items. Upon selling an item, the item will be removed from a player's inventory and their balance will be updated with the sell price.

User Story #30

As a user, I want to be able to buy items from the merchants between floors and sell items if necessary.

#	Description	Estimated Time	Owner
1	Create profiles of items including stats and effects	3 Hr	Michael
2	Create a function that displays profiles of items	1 Hr	Michael
3	Create a shop which holds items players can purchase	1 Hr	Michael
4	Create a function that displays the shop	1 Hr	Michael
5	Create a function that handles purchasing items from the shop	2 Hr	Michael
6	Create manual test cases	1 Hr	Michael

Completed:

The "!rpg shop" function displays all the items a player can purchase. The "!rpg item" function can be called to inspect the stats and get a description of a particular item. The "!rpg buy" function handles purchasing an item. Upon calling this function, the bot will add the item to the player's inventory and deduct the price of the item from their balance.

User Story #31

As a user, I want to be able to be able to know what different enemies do.

#	Description	Estimated Time	Owner
1	Create profiles of different enemies/monsters.	3 Hr	Arun

2	Create a function that shows all of the enemies/monster profiles, their stats, and abilities.	2 Hr	Arun
3	Create a tag to #2 function that orders the monster by rarity when displaying them.	2 Hr	Arun
4	Create a tag option to #2 that shows individual monsters displayed if asked for.	2 Hr	Arun
4	Create manual test cases	1 Hr	Arun

Completed:

When the user types the command !rpg enemies the bot will respond with a list of all enemies in the game with all their stats. The user can also sort by a specific enemy name by using the command !rpg <enemy>. Lastly the user can search by a rarity including Common, Rare, Epic and Legendary to get a list of all enemies in each of these rarities.

User Story #32

As a user, I want to be able to use skills and have the option of attacking, defending, healing, or fleeing the fight, when fighting monsters.

#	Description	Estimated Time	Owner
1	Create a function that prompts the user of the encounter with the enemy and the available options they have and what to type to choose said options.	1 Hr	Arun
2	Create a function that shows the different attack options for the player, including skills and attacks the enemy with the player's selected attack move.	1 Hr	Arun
3	Create a function that shows the defense skills and defends the enemy's attack if selected.	2 Hr	Arun
4	Create a function that heals the player if using an item or skill.	1 Hr	Arun

5	Create a function that flees the fight but if in a floor sets the to the beginning of the floor they were previously on.	2 Hr	Arun
---	--	------	------

Completed:

The options commands pop up everytime the monster attack phase ends and the player's turn has arrived. Depending on what the player chooses to enter as a command whether that is attack, defend, heal, or flee, the bot will perform the correct output. If "lrpg combat Attack" is entered the player will attack the first monster, if "lrpg combat Defend" is entered the player will gain 10 armor temporarily, if "lrpg combat Heal" is entered, the player will heal for 50 health capping at their health bar max, and if "lrpg combat Flee" is used the player will flee the dungeon restoring their health and armor and saving location, but losing their rewards for potentially clearing that floor.

User Story #33

As a user, I want to be able to enter a dungeon with my character and his/her items and skills.

#	Description	Estimated Time	Owner
1	Create a function that starts the dungeon dive and notifies the user that the quest is starting.	2 Hr	Arun
2	Create a function that randomized the monsters that spawn on each floor and level from a given set	4Hr	Konstantin
3	Create a function that randomizes the loot dropped	4 Hr	Konstantin
4	Create manual test cases	1 Hr	Konstantin

Completed:

If the user enters "lrpg dungeon <DUNGEON-NAME>" the selected dungeon-name for the dungeon will begin and notify the user as well as telling them which floor they are currently on. In the dungeon the monsters will slowly progressively get stronger each floor but will still be randomized for their level of strength. The dungeon will drop gold upon completion of each floor and bonus gold upon clearing the boss floor, this gold can be used to buy items.

User Story #34

As a user, I want to be able to enter a dungeon with my character and his/her items and skills.

#	Description	Estimated Time	Owner
1	Create a function that shows a leaderboard of the top 3 fastest completions of the dungeon run	3 Hr	Konstantin
2	Create a function that will write to a text file the record of the top three runs	3 Hr	Konstantin
3	Create a function that will overwrite one of the top three runs given that the most recent run is faster than one of those on the leaderboard	4Hr	Konstantin
4	Create manual test cases	1 Hr	Konstantin

Completed:

The leaderboard is displayed when “!rpg leaderboard” is entered. The bot will show the leaderboard for the fastest dungeon clear times and the user that cleared them in that time. These scores are updated on run if they are faster than the leaderboard runs.

User Story #35

As a user, I want to be able to see a leaderboard of the players who have won the most rounds, won the most money, and gotten the most blackjacks.

#	Description	Estimated Time	Owner
1	Create a function that obtains the players on the current server, sorts the players according to the criteria, and displays it to the user.	2 Hr	Matthew
2	Modify the player class to keep track of most won rounds, most money won, and most blackjacks.	1 Hr	Matthew
3	Create manual test cases	30 Min	Matthew

Completed:

For this user story, I added the 'leaderboard' argument to the '!blackjack' command that displays the blackjack leaderboard. I added additional fields to the Player object to keep track of the statistics and incremented these fields when the conditions are met, such as winning a round or getting a blackjack. Additionally, I created a function that parses these fields and constructs the message displays the leaderboard statistics. It also frequently checks the reactions of the message and edits the message if a user reacts with a reaction corresponding to another statistic. I also made the bot react with the keycap B, W, and M reactions to the message, which correspond with most blackjacks, most won rounds, and most money won.

User Story #36

As a user, I want to play a slot machine.

#	Description	Estimated Time	Owner
1	Create a function that directly messages the user a message representing the slot machine.	1 Hr	Matthew
2	Create a function that tracks the reactions of the user to the slot machine and runs the slot machine using random number generation.	1 Hr	Matthew
3	Create manual test cases	30 Min	Matthew

Completed:

For this user story, I created the '!slots' command that directly messages the user a message that resembles a slot machine upon calling the command. The message contains the three slots, the user's balance obtained from the Player object, the winning combinations and amounts, and the reactions corresponding to the bet amounts. I created a function that sends this direct message and monitors the reactions of the message to determine if the user made a bet to play the slot machine. Additionally, I made the bot react with the keycap 1, 2, 5, and 10 reactions to this direct message, which correspond to the different bet amounts. Upon the user reacting with any of those reactions, the function will run the slot machine by continuously editing the three slots in the direct message and displaying a newly generated random integer for each slot. After running the slots, the integers in the three slots of the direct message are obtained and the result is determined, adjusting the balance in the Player object accordingly.

What did not go well?

This sprint we ran into a similar issue as the previous one where everyone had their own clearly defined modules and user stories that did not overlap much. Because of this we did not really know the status of the modules we did not work on until we came together for the sprint review. This is most likely due to our meetings not being focused on the right topic to ensure everyone shares what they've done since the last meeting.

In addition, our sprint review went a bit overtime because we had a lot of new features to show off. There were too many acceptance criteria we had to show during the review which led to a long sprint review.

We had no incomplete user stories this sprint.

How should you improve?

One thing we could improve on if there was another sprint would be communication again. Similar to sprint 2, the work we did was mostly separate from each other so there was not much communication between team members on the progress of their modules. Most of the team did not know the progress of the other modules until it was time for the sprint review. Luckily there were no issues with conflicting packages this time.

Another thing we can improve upon is making more concise and specific acceptance criteria. While having thorough acceptance criteria is useful, we had many user stories with more than 3 acceptance criteria which gave us very little room to breathe with the time limits on the sprint reviews. Having more concise and fewer acceptance criteria would help us narrow down what really matters while also making our sprint review demos more organized and time efficient.