Dev Log

Day 1:

* Learned how to implement 3rd person camera system

Day 2:

* Learned the basics of C# and scripting
* Learned how to use GameObjects with vectors
* Created a basic player object with a controller
* Learned how to implement WASD movement script

Day 3:

* 3D modeled a skyscraper and an arena for use in unity
* Created a basic city scene in unity using the new models
* Scripted a system that changes the active scene if the player enters a building

Day 4:

* Scripted a fixed, top-down camera system for indoor scenes
* Tested lighting methods for indoor scenes

Day 5:

* Created a 3D model for the Gym that the player owns in the main city
* Added sprinting and jumping functionality to the movement script
* Created a new scene for the interior of the Gym

Day 6:

* Created a 3D model for our character and rigged it in blender
* Imported new character model to unity, and added it to our player object
* Tweaked settings for the camera system

Day 7:

* Created the “Interactables” script which allows the player to interact with specified objects

Day 8:

* Created a new scene for the Battle system
* Made a basic arena where the battle takes place
* Created objects for both the player and enemy monsters in battle
* Implemented a camera system that follows the player and enemy, keeping them both in frame as they move across the stage
* Created a modified version of the Movement script for 2D movement while in battle
* Created the “Moves” script which uses an Enum that contains all the moves a monster can use in battle, and the stats for each move (damage, accuracy, cost, type, range, etc.)
* Created the “Monster” script which contains all stats and variables used by the player and enemy monster in battle.
* Created “Player” script which inherits from “Monster”
* Created “Enemy” script, which is a modified version of “Player” that does not respond to user input. Also inherits from “Monster”
* Added new stat (energy) to “Monster” which is depleted when a monster uses a move, and recharges over time

Day 9:

* Modified “Enemy” script so the enemy will attack whenever its energy > 50
* Modified “Moves” script by splitting the single Enum into 3 Enums, each containing the moves that can be used for its corresponding range (close, mid, long)
* Implemented functionality to get the player’s location and its distance from the enemy
* Reimplemented range to now be the distance between the player and the enemy
* Created inheritance hierarchy [Moves]->[Monster]->[Player, Enemy, Battle]
* Implemented energy cost to perform an attack
* Implemented functionality so that different attacks are used at different ranges
* Target now takes damage if hit with an attack
* Created a primary and secondary move set that allows the player to customize which moves their monster can use in battle (3 moves per set, 1 move per range)
* Added functionality so the player can switch between the primary and secondary move sets by pressing “W” or “S”
* Created a “damage calculator” which calculates the total damage to be inflicted on the target. Uses stats from “Player” and “Enemy” to get final calculation

Day 10:

* Created a canvas for UI implementation
* Added a Health Bar Slider for the player and enemy
* Created “HitDetection” script to get data from “Player”, “Monster”, and “Enemy” and display it on the UI
* Implemented “hit or miss” system which calculates if an attack hits the target or not
* Added functionality so UI displays if an attack hit or missed
* Added feature so specified text variables disappear after a predetermined amount of time
* Added an Energy Bar Slider for the player and enemy
* Added feature where a successful attack displays the move used + damage dealt
* Added current range to UI display
* Added 3 buttons to the player and enemy sides of the UI, each button contains a move and its energy cost for that range
* Added a victory screen if enemy hp <= 0
* Added a defeat screen if player hp <= 0

Day 11:

* Designed new sprites for all UI systems
* Reimplemented all UI systems
* Fixed bugs with sliders
* Implemented a match timer that counts down from 60 seconds when the match starts. If both monsters are still alive at the end of the time, the monster with less hp loses   
  (or possibly based on # of successful hits, # of attacks dodged, etc.)
* Added feature so the name of both monsters are displayed on their respective sides
* Imported scorcher model and incorporated it into the Player object

Day 12:

* Finished modeling the player mesh
* Rigged the player model and started animations
* Created an Idle animation for when the player is not moving
* Created a (crude) running animation for when the player is in motion
* Created a 3D object for the capture crystal using blender
* Added textures to the capture crystal
* Created an animation for the capture crystal
* Imported the new items into the city scene
* Switched out the outdated player model for the newer version
* Created a new script “PlayerAnimation” that gets the players motion and returns it to an animator that changes the current animation depending on the return value
* Modified animator so the animation plays in any direction the player moves
* Implemented functionality so the walking animation speeds up while the player is sprinting
* Modified 3rd person camera by adding follow zoom and tweaking angles (it just feels better now)