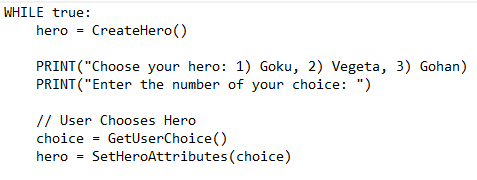
A diagram of a process

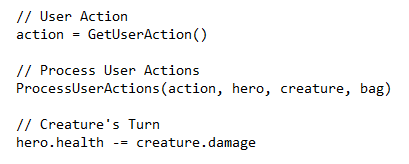
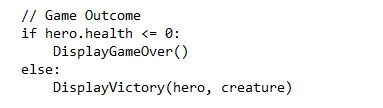
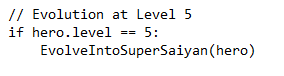
Description automatically generated

Now I will recreate the rpg game I had previously created while inheriting the imperative programming paradigm and using the programming language C. My high-level pseudocode for the game is as follows:

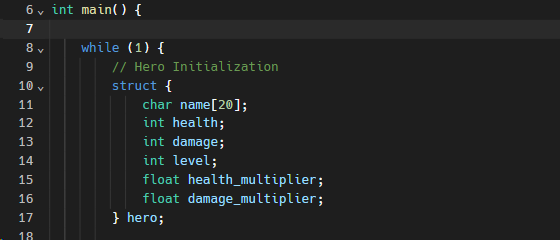
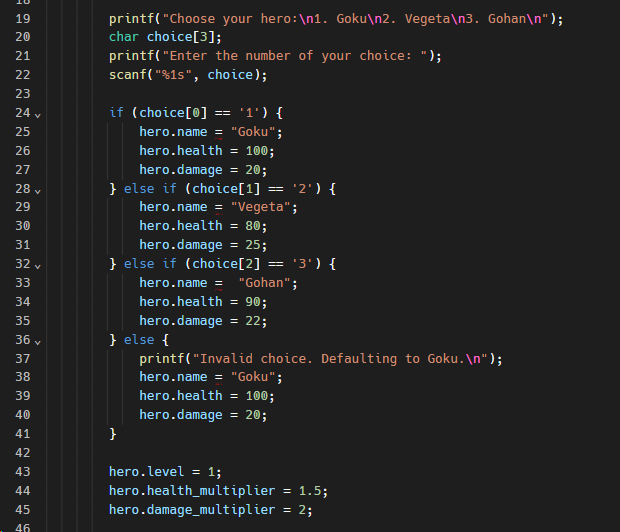
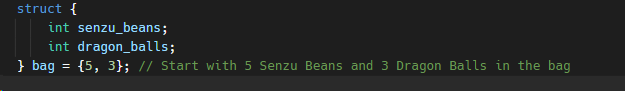
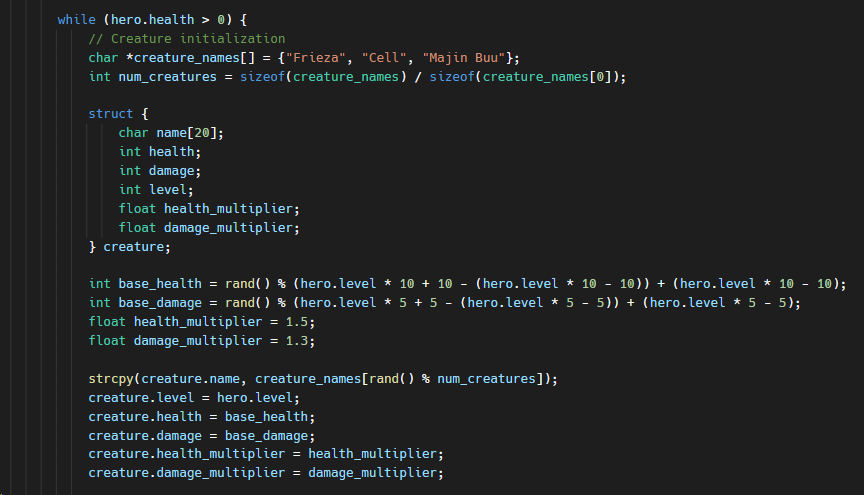
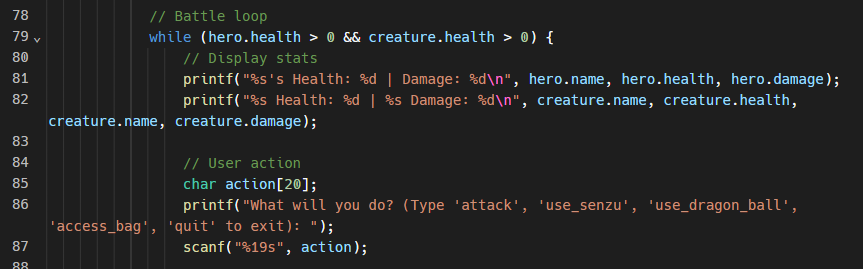
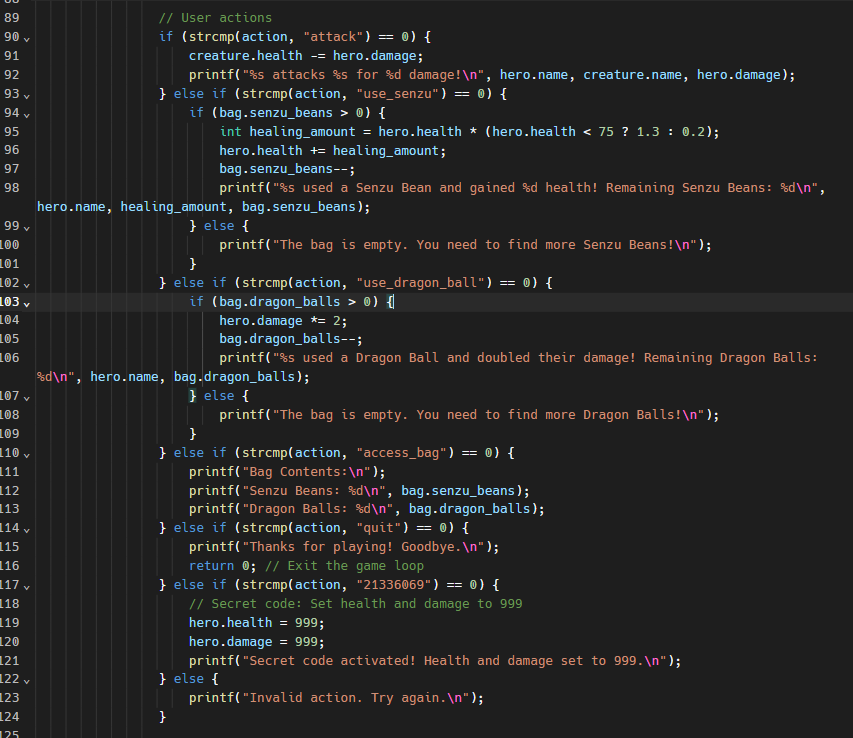
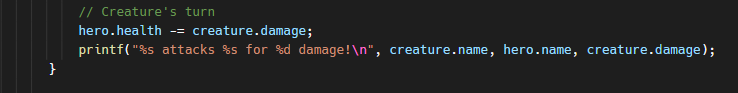
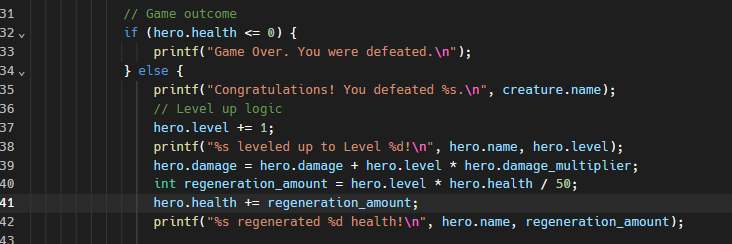
1. Just like the way I coded the RPG using procedural I will start off by initialising the hero however I will incorporate everything inside game loop to emphasise the imperative programming paradigm.  After the hero is initialised, the player will choose their hero by inputting a number between one and three which represents each hero. After this has been chosen the hero attributes will be assigned. EG level, damage, and health multipliers.
2. A black text on a white background

   Description automatically generatedThis bag structure contains the number of senzu beans and dragon balls inside the bag.
3. Inside the game loop the creature is initialised with their health, damage, health multiplier, and damage multiplier stats. The creatures’ stats are randomised and based off the hero’s level to ensure similar matchmaking to make the game fair.
4. A close-up of a computer screen

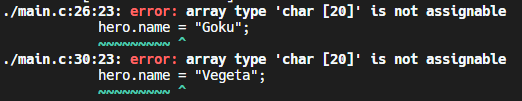
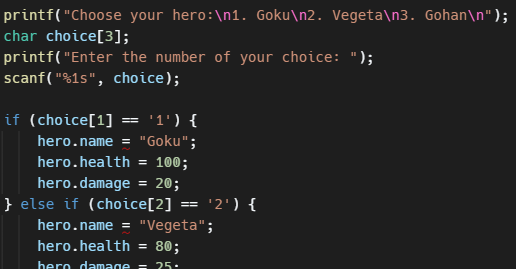
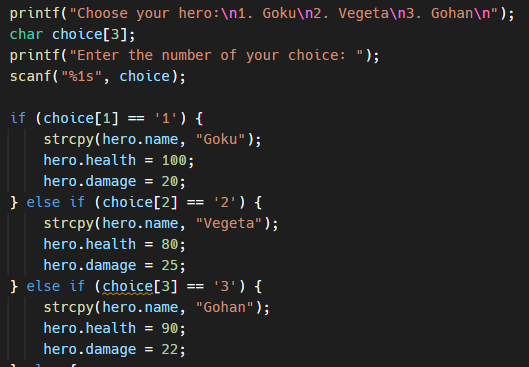
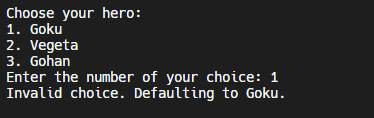
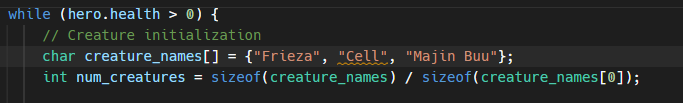
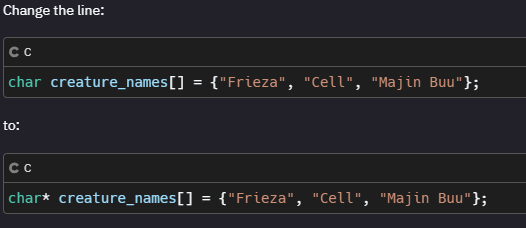
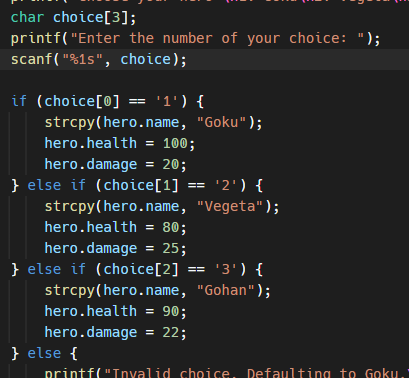
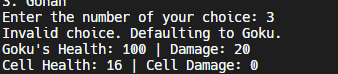
   Description automatically generatedInside this loop the battle will commence as the while loop will only run while the hero’s and creatures’ health are greater than 0. DisplayBattleStats will print the hero’s and creatures’ health and damage.
5. A text on a white background

   Description automatically generatedInside GetUserAction the player will be prompted to choose between attack, use senzu bean, use dragon ball, access bag, and quit. ProcessUserActions will process the action the player decides to choose. Creatures turn is where the creature attacks the hero during the creature’s turn. 
6. If the hero’s health goes below zero game over will be displayed as the hero has lost the game. Else the hero is still alive, and the creature is defeated then a victory message will be shown, and the loop will loop again until the game is over when the hero dies. 
7. If the hero defeats the creature the hero’s level will go up by one. And both hero and creatures’ stats will be boosted retrospectively.
8. If the hero’s level is 5 they will evolve resulting in their health and damage stat being doubled and a message will be outputted. 

Code Journal

1. I used the pseudocode to set an overall layout on how to code my program. So, I started off by creating the game loop and initialising my hero. I used struct as it allowed me to group the heroes stats/details under one name. 
2. After creature initialisation I set up the user input for choosing their hero and assigning each hero they’re individual stats such as health and damage and the stats that are the same across all heroes was hero level, health multiplier and damage multiplier as shown below. These numbers are directly copied from my RPG in python to ensure the game works the same across each language and paradigm.
3. I added the bag structure and inside it was the number of senzu beans and dragon balls inside the bag. The comments in the image below also further explain this.
4. I then coded the hero loop that checks to see if the hero is still alive. Then I initialised the creatures using the same structure as the hero. I copied the same stats for the creature as I did in my Python version of this RPG and then I assigned the creatures attributes to matching variables.
5. I coded the battle loop very similarly to my python variation as the logic is the same. The loop will keep running while both hero and creature are alive, and their stats will be displayed. After that I added the user input to take the action the user will decide to take. 
6. Instead of using functions and procedures that are called in procedural I added the player action procedures into the loop to reduce the level of abstraction and make the code flow more explicitly and line by line to help in line with the imperative programming paradigm. I copied all the logic from the python version of my RPG to ensure the game runs smoothly across each language and paradigm.
7. After coding the user action I coded the response to the user action which will be the creature attacking the user the code here simply shows this. 
8. Finally, I coded the game outcome which prints game over if the hero’s health is lower than 0 and the game will break out the loop. However, if the creature is defeated the hero will level up and the heroes damage will go up and health will regenerate. The regeneration logic is the same as the python version of my RPG.

Errors and debugs

1. The first error I faced was this this error points to the assigning of the hero’s name. this is what the code looks like. The problem here is that you are not allowed to directly assign a string to an array in C Therefore to resolve the error I used the strcpy function to copy the value into hero.name array. This is what my code now looks like 
2. Upon running the code, I faced an error where the game was not progressing to the next part after the character is chosen.  I realised the problem had to have come from the creature initialisation this is what it looked like. I did not understand why the code wasn’t working so I used replit’s built in AI to debug the code for me and I found out that the array declaration attempts to assign multiple strings directly, which is not allowed for char arrays. To fix this, I needed to initialise the “creature\_names” array properly with string literals at index positions. After using the built in AI’s help I fixed my code and the creatures started spawning.
3. The error here was that I was trying to access the array elements using incorrect indices. In C array indices start from 0. After attempting to fix this with the code here  this is how the program looked when ran The new problem is that now anything other than 1 is not accepted s character can only choose Goku and no one else. After some research I’ve realised it scans the character of input and choice[0] means the first character of input and char choice[3] is an array of size 3 that holds the user input since the user is only inputting a number I changed it to char choice[1]. Here is my refined code. 