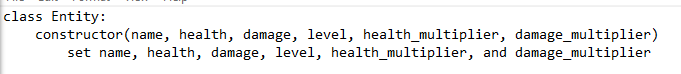
A diagram of a process

Description automatically generated

My final version of my RPG game will be programmed using the programming language C++ while inheriting the Object Orientated Programming paradigm. My high-level pseudocode for the game is as follows:

1. The entity class will be the base class for entities in the RPG. The constructor will initialise essential attributes.
2. The method attack will represent the entity’s action of attacking another entity. And reducing the targets health by the attacking entity’s damage. The level up method allows the entity to level up and increase the entity’s health and damage.A screenshot of a computer program

   Description automatically generated
3. The set health and set damage methods will allow for the entity’s health and damage to be changed. The heal method allows the entity to regain health during battle.A screenshot of a computer screen

   Description automatically generated
4. These functions will return the entity’s name, health, damage, and level.A screenshot of a computer code

   Description automatically generated
5. The hero class will inherit the entity classes attributes and behaviours.A black text on a white background

   Description automatically generated
6. The level up function will handle the levelling up process for heroes. When the hero levels up the level up function will be called from the base class to carry out levelling up operation. Also, there is a check to see if hero is level 5 to evolve into super Saiyan. The double damage function will be carried out when the user uses a dragon ball, and their damage will be doubled.A close-up of a text

   Description automatically generated A black text on a white background

   Description automatically generated
7. The creature class will inherit the entity classes attributes and behaviours.A black text on a white background

   Description automatically generated
8. The bag class will take senzu beans and dragon balls as parameters. These parameters will represent the number of senzu beans and dragon balls inside the bag. The use\_senzu and use\_dragon\_ball method checks if there are any available to use and carries out the effect of using either item. Also, the amount of senzu beans or dragon balls in the bag will decrement after either item has been used. The access contents method allows the player to view the amount of senzu beans and dragon balls currently inside the bag.A screenshot of a computer program

   Description automatically generated A black text on a white background

   Description automatically generated
9. The game loop sets up the structure for the main game loop where the user will be prompted to choose a hero. A screenshot of a computer

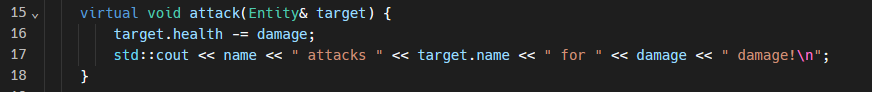
   Description automatically generated
10. The pseudocode here sets up the scenario where the hero will face a random enemy in a battle, and the details of the battle will be displayed on the console. The loop here will continue until the hero’s health drops to zero or below.A screenshot of a computer code

    Description automatically generated
11. The battle loop will continue as long as both hero and creature are alive. The hero and creatures’ current health and damage stats will be shown. After that the player will be prompted to choose an action, such as attacking, using items, accessing the bag, or quitting. After that the creature will attack the hero. If the hero defeats the creature a victory message will appear, and a new creature will spawn to battle the hero. If the hero is defeated, then the game will be over. If no one is defeated the loop will run again until a winner is concluded. A screenshot of a computer program

    Description automatically generated

**Code Journal**

1. The entity class I have coded is designed to represent the game entities with various attributes like health, damage, etc. The constructor initialises the attributes. A screen shot of a computer

   Description automatically generated
2. After coding the entity class I coded the attack method to allow the hero and creature to cause damage to each other.
3. I implemented the virtual level up method to handle entity levelling in the game. It increments the level by one, displays a console message with the entity’s name and new level, increases damage, and regenerates health using a multiplier based on hero’s level. I created a heal method to recover health by a specific amount and output the name and health amount healed. The double damage method I created doubles the heros damage and outputs this to the console.A screen shot of a computer program

   Description automatically generated
4. I implemented get methods to enable external code to retrieve information about an entity and the set methods allow external code to modify the health and damage attributes. This design helps facilitate controlled interaction with the state of an entity object. A screen shot of a computer program

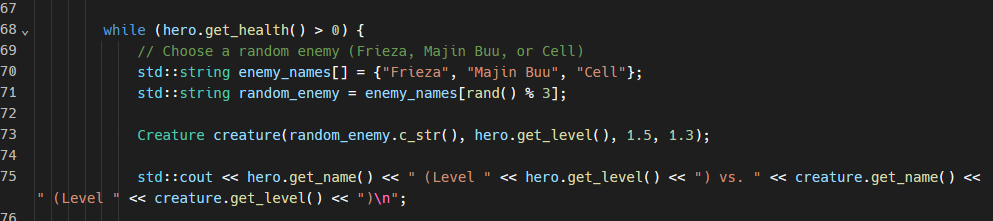
   Description automatically generated
5. In this part of the entity class, I defined the protected member variables. I used a character array to store the name of the entity, limiting the length to 20 characters. The health, damage and level variables store the entity’s health, damage and level retrospectively. I included floating point variables to the health and damage multipliers to allow dynamic adjustment as the entity levels up. A screen shot of a computer code

   Description automatically generated
6. I declared a class named Hero that publicly inherits from the entity class. The constructor initialises a Hero object with specified attributes by calling the constructor of the base class (entity) with the provided parameters. I overrode the level up method from the entity class to ensure that the specific behaviour for levelling up in the hero class is executed while retaining the functionality of the base class. The if statement checks of the hero is level 5. If true, it displays a message indicating that the hero has evolved into a Super Saiyan. Additionally, it doubles the hero’s health and damage using the set health and damage methods inherited from the entity class. A screen shot of a computer program

   Description automatically generated
7. I declared a class named Creature that publicly inherits from the entity class. The constructor initialises a Creature object with specified attributes by calling the constructor of the base class (entity) with the provided parameters. I used a formula that gives a random value for health and damage based on the creature’s level to provide variability in the creature’s health and damage. I then set these values to the creature’s health and damage.A screenshot of a computer program

   Description automatically generated
8. The constructor initialises the bag object with the provided counts of Senzu Beans and Dragon Balls. The use senzu function allows an entity to use a Senzu Bean. It checks if there are Senzu Beans in the bag, calculates the amount of healing base on the entity’s health, heals the entity, and updates the remaining Senzu Beans in the bag. The use Dragon Ball function allows a hero to use a dragon ball, doubling their damage. It checks if the entity isa hero using dynamic\_cast, doubles the damage if the entity is a hero, and updates the remaining dragon balls. The access content’s function displays the current contents of the bag.A screen shot of a computer program

   Description automatically generated
9. This section of my code prompts the user to choose a hero between Goku, Vegeta, and Gohan. The Hero object hero is initially set to Goku with specific initial attributes such as health, damage, level, health multiplier, and damage multiplier. However, if the player chooses Vegeta or Gohan the hero object is modified accordingly with the chosen hero’s attributes. A Bag object bag is created with 5 Senzu Beans and 3 Dragon Balls.A computer screen with text and images

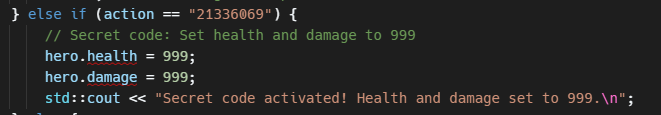
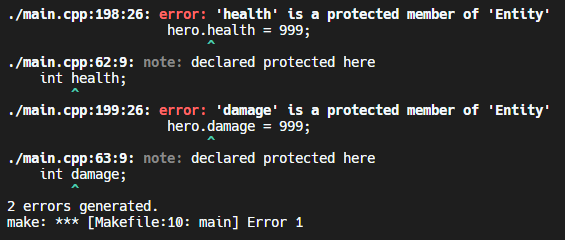
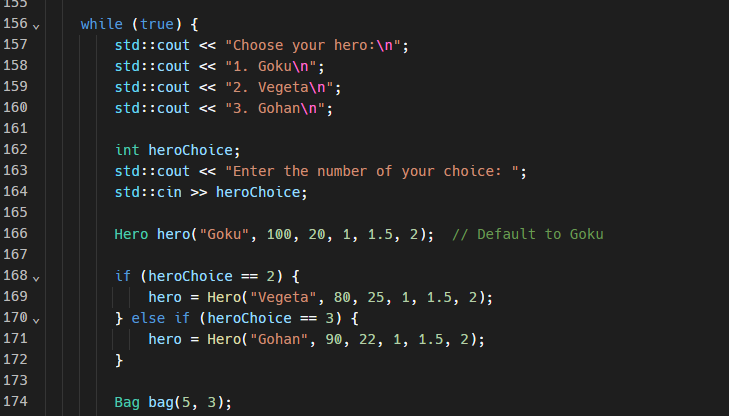
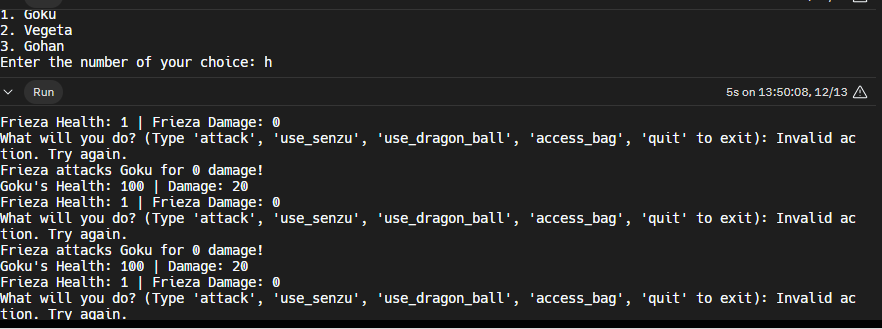
   Description automatically generated
10. The loop here continues if the hero’s health is greater than zero, meaning the player’s hero is still alive and able to continue the game. I used a string array called enemy\_names to contain the names of potential enemies (Freiza, Majin Buu, and Cell). I used rand operator to randomly select one of these enemies for the hero to face. I then initialised the creature with randomly chosen enemy name, hero’s level, and specified health and damage multipliers. The console displays both hero and creatures name and level son the player can see the matchup.
11. The loop here continues as long as both the hero and creature have health remaing to ensure that the battle continues until someone is defeated. Within this loop I displayed the hero and creatures health and damage so that the player can keep track of the battle and play their cards wisely. I then prompt the player to input their action so the player may decide what route they want to take during battle.A screen shot of a computer code

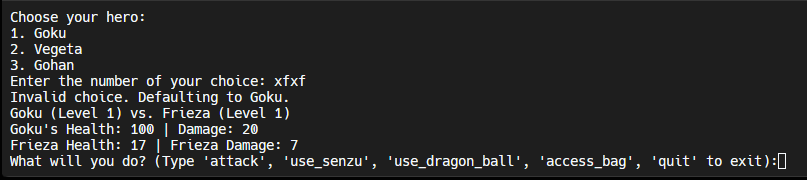
    Description automatically generated
12. Here are the simple if statements I have used to handle the players actions. Depending on the chosen action, various methods are called to perform corresponding actions, like attacking the enemy, healing the hero etc.A screen shot of a computer program

    Description automatically generated
13. After the player’s action the creature will attack the hero using the attack method. If the hero health falls below zero, then the game will be over. However, if the creature is defeated the level up method will be called and hero will engage in another battle.A screen shot of a computer

    Description automatically generated

**Errors**

1. The error I had here was that I was trying to set the hero’s health and damage directly, but I cannot do that because it is a protected member of entity. Therefore, I created set health and set damage as member functions of the entity class and declared them as virtual to allow for more ways to modify health and damage.
2. When the user inputs an invalid hero entry the game loops continuously. the code below shows how I have fixed and approached this error by clearing the input stream and ignoring the inalid input and automatically defaulting to goku when an invalid entry in inputted.A screen shot of a computer program

   Description automatically generated 
3. The problem I am faced with here is that the number of dragon balls are decrementing after 0. What’s supposed to happen is that a message should say there are no more dragon balls left. The problem here is I have failed to use the casting correctly. Therefore, I decided to change my code and make It work without the dynamic casting. A black background with white text

   Description automatically generatedA screen shot of a computer screen

   Description automatically generated What I did to fix this error is I made double damage a function in the entity class and overide it in the hero class. Now I can directly call the method without the need for dynamic casting.A black screen with white text

   Description automatically generated

