Read side by side with the Project Timeline Document this will explain the coding specifications for each checkpoint described

Coding component of The Legend of Densmore

Checkpoint 1:

- Implement the Point2D class from PA3 for the game's general movement
  - We will not use vectors since movement will solely depend on changing one component of the 2D point (i.e. the player will only move up, down, left, or right by one value. Each room lays one value away from the previous
- With or without the graphics the game should prompt the user which direction they want to move in.
  - If there is no room in the desired direction the game should make some error statement along the lines of "No room in the desired direction" and prompt the user for another input
  - There should be error messages if an invalid input is entered and the user should be prompted to give a valid input
  - Either at the beginning of the game or for each movement call the game should tell the user what the viable inputs are
    - W/w for up
    - S/s for down
    - D/d for right
    - A/a for left

        - Q/q to quit the game (will be removed when the program becomes a website)

- For the time being, the Boss room and Shop will be hard coded locations but will not do anything in order to ensure the player can move freely
- There will be a random chance in each room that x number of enemies appear. (we will discuss the actual percentages but for example if we get a random number between 0-15, no monsters, 16-50: 1 monster, 51-80: 2 monsters, 81-100 3 monsters
    - Have a monster counter implemented in order for the battle function to work
    - If a room either does not have a monster or has had its monsters defeated, the room should never contain monsters again (for example if the player moves through the room again)

Checkpoint 2:

- When a player enters a room with monster(s) they should be prompted with two options:
    - B/b will send the player back to the previous room but they will lose a percentage of their health as a consequence (maybe scale with the amount of enemies)
    - F/f will start the battling function (assume all subsequent bulletin points are for the battling function unless otherwise stated)
- A loop will occur that prompts the player to either attack or defend and then generate a random value for the actions of the enemies
    - Player attacks will have a base damage of ___ (figure out together)
    - There will be a percent chance for the player to deal a critical damage which will be 2x the base damage

    - Until we start making different level enemies, all enemies will have the same health and damage output (to be determined in the meeting)
    - Enemies can either defend or attack but it will be randomly generated
    - Once a player enters combat they are not allowed to run from the room
    - The player can only leave the room after all enemies are defeated and some victory statement should be displayed to the player
    - If the player loses all their health the game ends (an if statement should be placed at the top of the while loop for battling)
    - Inputs for battling are
        - A/a for attack
        - B/b for block (block is explained further in checkpoint 5)

Checkpoint 3:

- The damage upgrade should only appear after completing a room with enemies in it
    - The chances of it appearing are still up to chance, to be discussed how much it improves damage and how often it should appear
    - Increases base damage and in turn the crit damage
- The health upgrade can appear either in an empty room or after defeating an enemy (up for debate)
    - Theoretically can get a health upgrade after every enemy
    - The chances of dropping are random (need to discuss percent chance)
    - If the player is at full health when receiving the upgrade, increase their overall health by some amount, otherwise fully heal the player

Checkpoint 4:

- The win condition occurs if the player reaches the final room AND makes the boss lose all of its health (until different enemy types are implemented have the final room be hard coded to just have a single normal enemy
- The game should check after each turn of combat if the player has run out of health or not, if so, a game over message will occur
- If a player selects “play again” the entire program will reset
  - When multiple floor designs are made the floor which the player is on will be randomized

Checkpoint 5:

- We will discuss the health and damage of the enemies (and add them here) in general though:
  - Level 1 enemies should be relatively trivial dealing not a lot of damage and being able to be defeated in 1-2 turns
  - The boss should take 5+ turns (if the player has received damage upgrades)
    - Deal a sizeable amount of damage so that if a player at full health would still be able to win but barely
- A new base stat for the player should be implemented (maybe an upgrade as well but I don’t think we need it) that allows the player to avoid all damage by “dodging” the attack
  - Only the player will be able to dodge (enemies cannot dodge)

    - It is a set percentage however if an upgrade is implemented, it should boost the dodge state by a small percentage
- Blocking, as mentioned prior, will halve incoming damage in order to preserve the player's health
    - We should implement some reward for blocking otherwise blocking will serve next to no purpose, maybe increase the damage of the player's next attack by 1.5 (will not stack)
        - Can maybe include some text box like "Player retaliated"
- Enemies will randomly either block or attack the player

Checkpoint 6

While checkpoint 6 is only done if we have the time to do it, I believe the first point is very important in terms of making the game more fair but it could take time to implement

- Potentially scale the chances of finding higher level enemies the further a player is from the starting room
    - All adjacent rooms to the start would at most have 2 level 1 enemies for example
    - By the boss room have a higher chance of encountering level 3 enemies
- If time allows, the shop will be implemented into the game
    - Its location will be hard coded to be near the boss room
    - Will contain purchasable items that increase the player's stats
    - After each room (with enemies only) the player will receive 1 coin

  - The shop should scale the price of items so that by the end of the floor the player can choose to boost two stats (if we want, we can make it so that there is no currency and instead the room acts as like a “blessing” room
  - A message should appear when the player enters this room
- The boss key should most likely be hard coded or set to randomly appear within a room but the fact it appears is guaranteed that way in an unlucky playthrough a player isn’t forced to quit because the key never appeared

Fadi Section:

- Individual health bars for each enemy and probably a number to represent the enemy level
- Constant health bar for the player to know how much health they have left
- Option select for when a player wants to battle or run and if battling, if they want to attack or shield
- When upgrades are introduced implement a text box explaining what was upgraded
- Box to say if the player has died or if the player has beat the final boss and for each room