Project Bellum: Boredom Cure

Problem Statement:

From what my group has observed, people around the world and in the PSHS CAR Campus have experienced boredom often, and this boredom and lack of something to do has been related to worse mental health and wellbeing, as scholars we don't want to see other scholars and people around the world feel worse because of having nothing to do so we came up with Project Bellum: Boredom Cure for those scholars and people around the world who face boredom and experience sadness and result in a worse mental health well being. Proof of this is, from my group's observations, when my friends get sad or even irritated because of having nothing to do during long periods of free time where they have done all they can do and it is raining so playing outside is not an option. An example of this is when my friend Klydemar from 8-Camia once got sad during a day where we had nothing to do in class because it was raining and no requirements were need to be done but he had nothing to do so he felt empty and sad but that problem will no longer be encountered because of our group's game, Project Bellum: Boredom Cure

Project Objectives:

Our project's realistic objectives and goals is to help students around PSHS Car Campus be relieved of stress and free from boredom. We can say we have helped the students once we have gotten 5 students to test our game and approve of it as a good way to relieve stress and cure boredom. Our project's output would be a text-based RPG game made in Python which is the game that will help relieve stress and cure boredom.

Planned Features:

1. In our project, the player will be able to roam around the world and discover new places and areas that are possible to go to and even interact with the NPCs and features those places have to offer, this is a common feature in most RPG games that helps players have fun and give them free will to do what they would like for that time.

2. In our project, the player will be able to encounter enemies that they can fight to win or lose against, these enemies in the game are called Umbras(creatures) which are the normal difficulty enemies meaning they drop less loot but are easier to fight and the Ultraluxes(bosses) which are bosses which drop more loot but have higher risk. The loot that enemies can drop will be souls, the currency in our project, learn new crafting recipes for better gear, and collect trophies by defeating different bosses which drop different trophies. This part of our project helps players relieve stress and cure boredom by releasing their stress through fighting the bosses and this gives them something to do in their free time, solving the boredom problem.

Planned Inputs and Outputs:

1. In our project, the player will be inputting text only and the system will output text only too with occasional ascii art. The player will be inputting their character name before the game and the system will output the name they give before the game as the name they will be referred to for the rest of the game.
2. In our project, for the roaming feature, the player will input if they want to advance or go back in our world with the player typing Advance and Back for this feature and as they roam they may or may not encounter random events. The system on the other hand will output what the player has done and the result of that action then giving the player options again on how they plan to respond to the situation.
3. In our project, the player will find special POIs where they can do things in maybe a shop or just an eventful place and the system will give them options on how they would like to interact with the place then the system will show what the player has done and the result of their actions, then giving them the options again until the player wants to get out of the POI.
4. In our project, for the battling of enemies, when the player does encounter them the system will ask the player how they want to respond with a set of options like 1. Retreat 2. Fight 3. Heal but this set of options is not yet complete for our project. Depending on what they do and the player's level, gear, level of the enemy, the player can either lose or win against the enemy. The system will then say what the player has done and the result of their actions, it will also give the player a set of options like mentioned earlier with it showing the health of both player and enemy too.

Logic Plan:

1. Behind the scenes of our project in its code, the game will start with possibly a story telling part as an intro with print() statements.
2. The game will then start with the system asking the player if they want to advance with no back option yet because they just started, in the code there will be an if else statement that will continually update as the player progresses and moves forward or backward in the game
3. As the player progresses in the game, they will encounter enemies and POIs where they can choose to interact with or not, if they don't they continue to advance or go back depending on what they want to do. If they player does choose to interact with the enemy or POI, it is different, for the enemy, there will be print() statements showing both enemy and player's status and what happens as they battle with variables in the code keeping track of those statuses and conditions that happen as they fight, if the player wins there will be a random selection from a list of possible loot drops for the particular enemy they fight but if the player loses the code ends as the player loses the game. For POIs, the player will just have the option to go forward and backward as it is similar to exploring and roaming around the regular world then once they find an interactable area the system shall use print() statements to show the possible things to do and once the player inputs what they want to do the system will respond with print() statements on what resulted from their actions then going back to printing if they want to advance or go back.

4. When the player fights all bosses, gains all the gears and trophies, and basically completes the game, the system shall use print() statements to show the stats the variables that were continuously being updated have been stored with and then the system will congratulate them and finish.