Analysis: Limitations of the Proposed Solution

For my solution I have chosen a two-dimensional platformer with science themes- in less words, a game. I have decided to create this game using Java and while I am not too familiar with the use of object-orientated programming (OOP), for a game that is made up of objects interaction on a screen, the intrinsic nature of OOP should compensate for my lack of knowledge by being better suited for the task.

While careful planning and optimism are important features of any task, by applying the computational approach to a task you have to consider computation. This can be split into two sections in categories of this task. The first category is what cannot be turned into a computational solution. This could be due to limitations of the resources I have (I cannot run weather simulations on a school computer, as the hardware cannot run trillions of floating-point calculations per second) or a limitation of computing (I cannot ask any computer to find a 'good song', computers cannot handle abstract requests like one that requests a personal preference). The second category is what the programing team cannot compute. An example of this was the NHS database, the team was over their heads and the project didn't go well. As the sole person doing every aspect of this project with deadlines, I have personal limitations that become the limitations of the solution.

With that introduction explaining limitations and the relevance to this project, I will now list the limitations of my proposed solution:

- The first limitation is the scale of content of the solution (what may be referred to as a game, because that is what it is). With a single person both designing, programming and implementing features, it is important to figure out how long I can afford to make the game. I need to find a point of maximum user interest for minimal in game content (in terms of levels/stages). This means that while a bigger platformer generally will leave a bigger impression (mental rewards for overcoming obstacles), the solution will have to be of a limited physical size.
- The next limitation is complexity. Every time I add something to this game, the complexity increases. For example, if I add platforms, I have to then update the character classes and all existing environmental classes. It will become a problem to manage a project that is very complex. This also means that the memory used will begin to increase by a factor for every class implemented. As I am saving this on a school computer, the memory storage I have is also shared by other files, including the rest of the project.
- Interestingly, another limitation of the project would be the visuals. As I am creating all the graphics, as I can't achieve stunning photorealistic graphics, I'll have to use a consistent stylised approach to graphics to compensate for my limited artistic talents.
- Finally, I have my own limits. This means that I will have to really rely modular approach so that I can simplify each section if needed, meaning that my lack of experience will only affect certain aspects, rather than the whole game. An example of this would be splitting the character's movements from its interactions. This means that in the event that I have to compensate for an interaction, the core aspects of the character's movement would not suffer in anyway.