Assignment

Problem Statement: *Building a floor planner application in python*

With the recent advent of VR capabilities, e-commerce is now capable of evolving the overall buying experience of a consumer by having an interactive viewer where any user can place an object of their choice in a virtual room setting. Since a vast majority of the e-commerce giants out there have a primary database of 2D catalogue (and not 3D meshes), it would make more sense that the user is able to ingest and interact with 2D object catalogue instead of 3D.

As part of this assignment, you have to build one such viewer *purely in python* (no frontend knowledge required). The detailed tasks are listed below.

Feel free to write at pulkitgupta@avataar.ai for any clarifications.

Tasks:

The assignment has three deliverables:

1. Build an interactive viewer where there is a 2D scene image as the backdrop and a 2D object image as the foreground object. One possible example of how the output should look like. Few things that you might need as input are present here. For this task, the bare minimum functionality is automatic rescaling of the object as it is moved around the scene.
2. After the above task, you will notice that the main object is always overlaid on top of other objects, for a true experience, occlusion awareness needs to be implemented in the object's motion.
3. The object still goes inside other objects. Ensure that the object always traverses the visible floor and does not go inside the scene objects.
4. **[Optional][Bonus]** Take a text prompt from the user and based on that change the background scene image as per user's text prompt. For this, you can refer [1-2]. In the experience that you built as part of tasks 1-3, you might be using the depth map. Therefore, for the scene image to be generated using the user's text prompt, use the original depth map as a conditioning prior so that the scene depth map does not change.

Note: All tasks need to be achieved programmatically (using python).

Deliverables:

As part of the submission, provide a link of the github repository with the code and a readme file describing the execution of the code, the thought process and the visual results. Also analyse where your approach works well, where it fails, ideas on what would be done to fix it etc. and include this in your report.

References:

[1] Stable Diffusion

[2] ControlNet