Project Report: Dream Villa – PS/CSC/22/0049

1. Project Overview

Dream Villa is a high-end, interactive Virtual Reality (VR) experience built using A-Frame (v1.4.2) and Three.js. The application presents a modern luxury villa with realistic textures, detailed architectural modeling, and interactive smart-home features such as light toggles and animated environmental elements.

The environment extends beyond the house to include a road system, street lamps, a basketball court, and a functional swimming pool, creating a believable neighborhood context and preventing the user from feeling isolated in an empty 3D space.

---

2. Design Choices

2.1 Architectural Concept

- Modern Open-Plan Living
  The villa follows a minimalist modern-house aesthetic, characterized by open-plan layouts, large glass surfaces, and a high-contrast material palette. Dark wood is contrasted with bright white tiles to emphasize clean lines and a premium feel.
  Large glass spans are simulated through A-Frame entities with transparent materials, enhancing views across interior and exterior spaces and reinforcing the sense of openness.
- Environmental Layering
  To avoid the "floating in a void" problem common in VR demos, the villa is embedded in a richer world. Surrounding elements—roads, streetlights, a basketball court, and the pool—provide scale cues and context.
  This environmental layering strengthens immersion and supports natural navigation, helping users understand where they are within a larger, coherent setting.

2.2 Asset Selection

- GLTF/GLB as Primary Format
  All 3D assets are imported as GLB to ensure:
  - Efficient loading and rendering in the browser.
  - Native support for PBR (Physically Based Rendering) materials, preserving realistic metalness, roughness, and normal-map details.
- Texture Mapping Strategy
  The project combines:
  - Local textures for reliability and consistent loading across deployments.
  - High-quality PBR normal maps, especially for the pool water, to create convincing highlights, reflections, and surface "sparkle" without heavy geometry.

2.3 Interaction Model

- Kinematic Rig for the Player
  A kinematic-body is used for the user rig to ensure proper collision handling and gravity, preventing the player from sinking through the floor or flying.
    - look-controls provide natural head tracking in VR, keeping the experience intuitive for both desktop and headset users.
      This combination results in grounded, realistic movement while retaining smooth first-person exploration.

---

3. Technical Challenges and Solutions

Challenge A: File Path Inconsistency (Local vs Deployment)

- Issue
  The scene worked correctly on a local XAMPP server but failed on Vercel/GitHub, returning 404 Not Found errors for models and textures.
- Cause
    - Windows (local) is case-insensitive, while Linux-based cloud hosting is case-sensitive.
    - Mismatches like **models/** vs **Models/**, plus leading slashes (e.g., **/assets/...**) caused broken asset references.
- Solution
    - Standardized all asset paths to match exact folder names and capitalization.
    - Removed leading slashes, switching to relative paths (e.g., **assets/Models/villa.glb**).
      After these corrections, assets loaded consistently across local and cloud environments.

Challenge B: Collision and Navigation ("Solid" House Problem)

- Issue
  Using **static-body="shape: mesh"** on the full house GLTF created a single, solid collision volume. The villa became an impenetrable block, blocking entry through doors and windows.
- Solution
    - Removed complex mesh-based collisions from the main GLTF model.
    - Added invisible **a-plane** entities with **static-body** to define walkable floor surfaces.
    - Created an invisible diagonal ramp over the stairs so the kinematic rig could "walk" up to the second floor without snagging on step geometry.

This approach simplified physics, improved performance, and restored natural navigation through the house.

Challenge C: Texture Syntax and WebGL Errors

- Issue
  The browser console showed “Unknown property” warnings, and some textures failed to render correctly. In some cases, invalid syntax contributed to WebGL instability or crashes.
- Solution
  - Carefully audited A-Frame component syntax and removed CSS-style comments placed inside HTML attributes, which are not allowed in this context.
  - Corrected **crossorigin** settings on external textures and models to comply with browser security policies (CORS), ensuring that external assets could be fetched and used safely. This cleanup stabilized the scene and restored reliable texture mapping.

Challenge D: Pool Animation Performance

- Issue
  Directly animating a high-polygon water mesh caused performance drops and stuttering, especially on lower-end hardware.
- Solution
  - Implemented a custom **pool-animation** A-Frame component that animates texture offsets instead of geometry.
  - By shifting the Normal Map and Diffuse Map at slightly different speeds and directions, a believable water ripple and flow effect was achieved.
  - Because only UV coordinates are updated, this method delivers smooth, realistic animation with virtually no performance penalty.

---

4. Future Improvements

1. Dynamic Lighting System (Day/Night Cycle)
   - Integrate an **a-sun-sky**-based system to simulate time-of-day changes.
   - Gradually adjust directional light intensity, sky color, and ambient light.
   - Enhance realism inside the villa, especially how chandeliers and interior reflections respond to changing outdoor light.
2. Advanced Collision Mesh with NavMesh

- Replace invisible ramps with a Navigation Mesh (NavMesh) to define walkable surfaces more precisely.
- This would enable more advanced locomotion options such as:
  - Smooth navigation around complex geometry.
  - "Click-to-teleport" features for users on mobile VR or users prone to motion sickness.

3. Spatial Audio 2.0 (Acoustic Dampening)
   - Currently, the jet object uses positional audio.
   - Future enhancements include acoustic dampening, where walls and doors attenuate external sounds.
   - For example, the jet engine would sound loud outdoors, then become muffled or quieter once the user enters the house, better mimicking real-world acoustics.
4. Interactive Furniture Customization
   - Introduce an in-scene UI that allows users to select furniture items (e.g., couch, chairs) and cycle through fabric colors and textures in real time.
   - Use dynamic updates to the **material** component to swap textures without reloading models.
   - This turns the villa into both a smart home demo and an interactive interior design tool.

---

5. Tools and Technologies Used

- Framework: A-Frame 1.4.2
- Rendering Engine: Three.js
- Physics: A-Frame Physics System powered by CANNON.js
- Assets: GLTF/GLB (primary), with OBJ/MTL support where needed
- Deployment Stack: Git for version control, GitHub for repository hosting, and Vercel for live web deployment

This combination of tools and design decisions results in a performant, immersive VR experience that balances visual fidelity, interaction, and accessibility across web platforms.