Documentation: PCPartVisualizer
Group: True Lab
Group Members: 22i-0878, 22i-0983, 22i-1321

Project Overview

PCPartVisualizer is a tool inspired by PCPartPicker that extends its functionality by allowing users to visualize their selected PC components and builds in a 3D environment. This feature-rich application combines hardware compatibility checking with an immersive 3D representation of builds, bridging the gap between functionality and user experience.

Scope

The project aims to:

- Facilitate 3D visualization of selected PC parts, helping users better understand their builds.
- Provide a seamless user experience by integrating cutting-edge frontend, backend, and rendering technologies.
- Maintain compatibility checking and build optimization, similar to PCPartPicker.

Reason to Exist

While PCPartPicker excels in listing, comparing, and verifying part compatibility, it lacks an intuitive visual representation of how components physically fit together. PCPartVisualizer addresses this gap, offering users:

- Enhanced decision-making with visual confirmation.
- Educational benefits for new builders learning about component layouts.
- An innovative way to explore and showcase builds in 3D.

Practicality

- User Base: Ideal for PC enthusiasts, gamers, and DIY builders.
- Accessibility: Easy-to-use interface for all experience levels.
- Real-World Use: Helps avoid assembly issues by visualizing fit and compatibility beforehand.

Legalities

- Intellectual Property: Any reference to PCPartPicker must respect copyright and trademark laws.
- 3D Models: Ensure all component models are sourced or created with proper permissions.
- Data Compliance: Adhere to GDPR, CCPA, or relevant regional data privacy laws when handling user data.

Licenses

- Frontend (Next.js): MIT License.
- Backend (Spring Boot API): Apache License 2.0.
- Database (Supabase Postgres): Open-source under PostgreSQL License.
- 3D Engine (LWJGL-based OpenGL): Open-source with BSD-style license.

How to Use

1. Select Components: Use the web interface (Next.js) to choose PC parts.
2. Visualize Build: The 3D engine renders selected components, showing their placement and fit.
3. Save/Modify Builds: Save your configurations for future adjustments.
4. Export Options: Generate part lists and visual snapshots of the 3D build.

Key Features and Recommendations

- Custom 3D Engine: Built with LWJGL, offering efficient and detailed rendering of parts.
- Frontend: Responsive, intuitive interface powered by Next.js.
- Backend: Scalable API in Spring Boot ensures smooth data handling.
- Database: Supabase Postgres provides reliable and robust storage for parts and build data.

Recommendations:

- Regularly update the 3D parts library to match new releases.
- Incorporate VR/AR support for next-level user experience.
- Implement user feedback mechanisms for continual improvement.