A SMART CAR MODEL EDUCATION AND SOCIAL COMMUNICATION PLATFORM IN THE AUTOMOTIVE INDUSTRY USING VIRTUAL REALITY AND ARTIFICIAL INTELLIGENCE

David Lee [1], Moddwyn Andaya [2]

[1] Arcadia High School, 180 Campus Dr, Arcadia, CA 91006
[2] Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

The program's key technologies include the AI program which helps the user create a description for their models, a walking system for the user to navigate the map, and a model importer system to import the models of the user in order for them to view it. I addressed design challenges by adding many unique and specific words to the AI in order for it to come up with distinct descriptions for the user [2]. The walking system was made to be smooth so that the user can have the most realistic experience in the program. The model importer only imports the model in a set location so that it would always be viewable by the user in that location. The design challenges were also addressed by looking at inspiration online for features to add and also different models that can be added in order for the app to become more realistic.

KEYWORDS

AI-generated descriptions, 3D model exploration, Realistic navigation, Model import system

1. INTRODUCTION

The problems that influenced me to create this project was the fact that many car designers in the industry lack an app to see their own designs for free, which may be very beneficial for inspection of a model before production. According to flyingshapes.com, "Virtual reality design environments like flyingshapes carry the promise of transformation" [1]. Many of these apps exist, but I believe that most of these apps require payments, and sometimes have very extensive and complex model importing processes, which would make it harder for an amateur to understand. Many of these apps also have UIs which are hard to navigate, and sometimes you cannot customize these apps to your liking. With my app, I plan to make the background changeable and allow the user to set keybinds based on what controls they would want. These apps usually are made for industry professionals, and therefore are only accessible to them. However, I am making my app accessible to not only industry professionals, but also casual users who want to see their models. I plan to implement features that can benefit both casual users and people who actually need it in their business. I believe that doing this will ensure more customers who will be using this app in the future.

Section 5 reviewed methodologies that each tackled some of the important aspects of 3D modeling, AI in design, and usability in complex systems. The first methodology was about usability factors in 3D modeling for VR education [11]. It focused on enhancing the learning

David C. Wyld et al. (Eds): ADCO, CMCA, NeCoM, NLPI, CSITEC, BDBS, DMML, SP – 2025
pp. 97-105, 2025. CS & IT - CSCP 2025 DOI: 10.5121/csit.2025.150508

experience through interactivity and dynamic compatibility but lacked features like content generation and advanced customization options. The second methodology studied AI in design, emphasizing efficiency and creativity through automation. While impactful, it didn't provide immersive visualization or specific applications that include automotive design. The third methodology provided usability principles for 3D parametric tools, targeting professional architects and engineers without considering casual users or integration with AI [3].

Our project enhances these methodologies by combining their strong points: it provides a virtual showroom for immersive visualization, integrates AI to generate car descriptions, and offers user-friendly customization. These features ensure accessibility for both professionals and beginners alike, filling the gaps in functionality and inclusivity.

The idea behind my app is for it to have a changeable, realistic background, an easily navigable interface, AI support to help create descriptions of models, and the feature of importing models from the user's computer. I plan to use ChatGPT for the AI, because it is a reliable and proven AI system that has benefited many users in the past [4]. I need a changeable background because I would like the user to have customization over his app, therefore allowing them to edit which background it uses. For my first background I made a showroom with a street in the background to show how the car would look if it was inside of a dealership. I will use free assets in order to put together the background, and I plan to add moving traffic in the background to make this scene more realistic. The AI moving traffic will be moving around in the background while the model is displayed, and they will despawn and respawn outside of the view of the user. Another feature I plan to add is the ability to change the scaling of the model, in case the user does not like the standard scaling of the model inside the game.

During the experiment, I have listened to the feedback users have provided for this app. This includes the lack of variety of AI and the emptiness of the map. They have complained about the AI seeming robotic sometimes and not having the best descriptions, which can be fixed by adding more keywords into the AI [5]. The complaints about the map being empty can be fixed by adding more props and moving vehicles into the map, thus making it feel more lively. The interior of the building will also be more detailed later on, and include the second floor details in order for the user to not feel like the game is lacking features. I think this experiment was an overall success in determining what I needed to fix in the program, and I will try to fix every feedback that the users gave, including the AI and the emptiness of the map.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. GLTF

Sample Skeptical Question: What if I have a model that is not GLTF?

Sample Response: You can convert the model into GLTF in blender or another software or website that allows conversion of a 3D model compatible file extension, or else it cannot get imported into the app as it is required [12]. If downloading models from external websites that allows you to download 3D models, they can sometimes have an option to download different file extensions of the model or simply search for models that are GLTF by default.

2.2. AI

Sample Skeptical Question: What is the AI useful for?

Sample Response: The AI is useful for creating descriptions of the car. The AI can tell us what to model into the car, which may help the designer have more originality. The AI describes things such as the shapes of the car or the features of the car itself. This ultimately saves time because many designers often spend a lot of time trying to find original ideas of what kind of car to design, which makes them lose the time they might have for other activities.

2.3. Showroom

Sample Skeptical Question: What can you do in the showroom level?

Sample Response: You can walk around the showroom, but in the future I plan to have openable doors that lead to the outside. The outside will have an invisible wall surrounding the area, so that the User cannot walk on to the road. The main purpose of the outside is for the user to see their model from the window outside. There is a second floor that is currently empty, and you can look around with your camera while being inside the showroom. The second floor is planned to have even more space for the user's models, and can feature some decorations which make the environment look better.

3. SOLUTION

When you open the app, the first menu you will encounter is the model importer panel. This panel contains buttons that allow you to import glTF format 3D models [13]. There's two ways to import models, one is pressing the red "Load ZIP" button allowing you load glTF project files from a zip which the app automatically unpacks. And two, the yellow "Load Folder" button that also loads glTF project files straight from a folder. Once project files are located and when you press the green "Load Model" button underneath, there are a couple bars that will show change. First is the path bar that shows the path of your model files, second is the status bar which shows the current importing status messages whether that be an error or successful import message. And last is a simple loading bar to show importing progress. After the progress is finished, with the help of the UnityGLTF package, the app will unpack the model files and spawn the model into the showroom scene including its textures, closing the main panel allowing you to walk around the scene [6]. By pressing Q by default, you can again open up the main menu panel. Another menu included is the AI generator menu which can be accessed by a blue button labeled "GO TO AI GENERATOR" located at the very bottom of the screen. This will take you to another menu with only one green button labeled "Generate" that will generate a car description for 3D car modeling and design inspiration. And anytime again, pressing Q or pressing the blue button located at the bottom screen of the AI Content menu labeled "GO TO MODEL LOADER" will bring up the main panel.

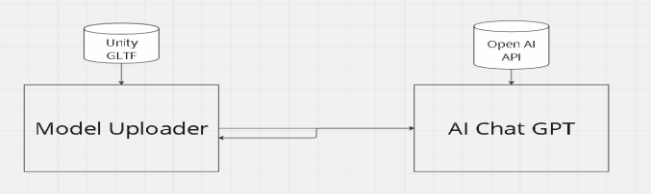

Figure 1. Overview of the solution

The system's first core component is the glTF loader that loads 3D models that have a .gltf or .glb file formats which Unity doesn't natively support [7]. The loader itself also allows textures to be imported contained in the same project files folder fully showing its final model. The main purpose of this loader for this app is to import car models and preview them in the showroom.

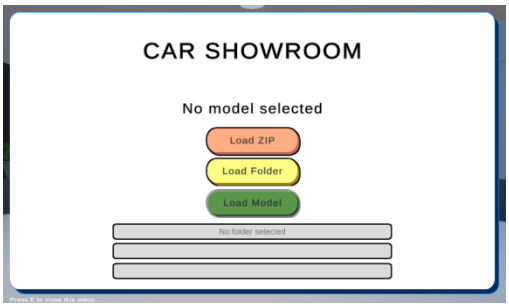

Figure 2. Screenshot of the showroom

```
ImportOptions importOptions = new ImportOptions();
GLTFSceneImporter sceneImporter = new GLTFSceneImporter(uriPath, importOptions);

statusText = $"Loading model: {gltfFileName}...";
print(statusText);

try
{
    await sceneImporter.LoadSceneAsync(
        sceneIndex: -1,
        showSceneObj: true,
        onLoadComplete: (model, _) =>
        {
            if (loadedModel != null)
            {
                Destroy(loadedModel); // Clean up any previously loaded model
            }
            loadedModel = model;

            statusText = $"Model {gltfFileName} loaded successfully.";
            OnLoadSuccessful?.Invoke();
            print(statusText);

            AdjustModelTransform();
        },
        cancellationToken: default,
        progress: progress
    );
}
catch (Exception ex)
{
    statusText = $"Failed to load model: {ex.Message}";
    print(statusText);
}
```

Figure 3. Screenshot of code 1

The glTF model loader first requires us to use the UnityGLTF package to access the GLTFSceneImporter class that processes all the models, scene, and texture files and is ready to be accessed and controlled in Unity. The class requires a glTF file name where we then input in the uri path of the project files. This path is made available and filled out before loading in the model when we import a zip or folder of the glTF project files. After this, we do a try and catch to see if we are able to import the glTF project as not all features are implemented and may not be supported in Unity yet. If successful, the scene importer loads the project files with the "LoadSceneAsync" function that comes with a onLoadComplete action that returns a model after import success. This model is essentially the final 3D model that we have imported and now control freely in Unity. If any previous project is loaded in the showroom, then that will be first destroyed and replaced with the new one. Once the project model is loaded then we call the function "AdjustModelTransform" that completes a couple things. First is the scaling of the model that changes the local scale of any model to perfectly fit inside a specific size bounding box that fits within the showroom scene. And then it also moves and rotates the object right on top of a specified location we set within the scene for viewing.

The app's second core component is the AI content generator that generates car descriptions made for 3D artists and designers to help and inspire to create new car models that they can make. This system mainly uses the OpenAI package for Unity that allows us to access the API of the most famous ChatGPT service.

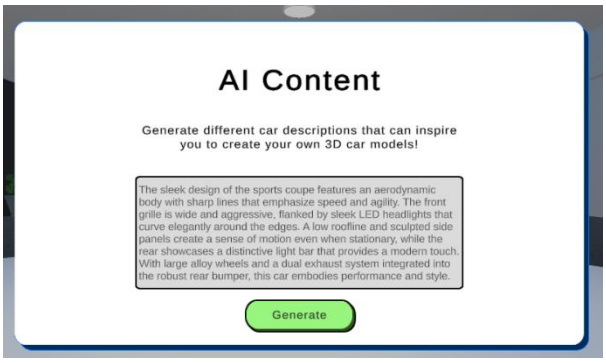

Figure 4. Screenshot of the AI Content

```
void Start()
{
    messages = new List<Message>
    {
        new Message(Role.System, instructions), // Starting prompt
    };

    openAI = new OpenAIClient();
}

0 references | 1 reference from Unity
public void Generate()
{
    if(generating == false)
        SubmitChat(".");
}

1 reference
async void SubmitChat(string input)
{
    if (string.IsNullOrWhiteSpace(input)) { return; }

    generating = true;
    outputText.text = "";
    loadingText.SetActive(true);

    var chatRequest = new ChatRequest(
        messages,
        Model.GPT4oMini
    );

    var response = await openAI.ChatEndpoint.GetCompletionAsync(chatRequest);
    string output = response.FirstChoice;

    outputText.text = output;
    loadingText.SetActive(false);
    generating = false;
}
```

Figure 5. Screenshot of code 2

To access the OpenAI API, the code first needs to access the OpenAIClient which is our main port of access to the backend services [14]. At the start of the app, we create a list of Messages which contains data about the message itself that is being sent and who sent it (also known as Role). Along with the new list of Messages at the start, is the addition of a new Message sending the instructions (the prompt written to describe what output we want the AI to generate, which in this case, car descriptions). This message is sent to the system Role to be its initial full

conversation prompt. The “Generate” function is connected to the UI button on the AI Content menu panel within the app that simply calls the function “SubmitChat”. This function contacts the OpenAI client by calling “openAI.ChatEndpoint.GetCompletionAsync” that calls the backend chat service, and has the code wait asynchronously to receive the AI output. This output, called by writing “response.FirstChoice”, is then sent back into a text box on the app.

The final core component of the app is the Player Controller. The showroom app requires us to view the actual import glTF project model so we need to have a player that walks around showroom level so we can view the car model all around.

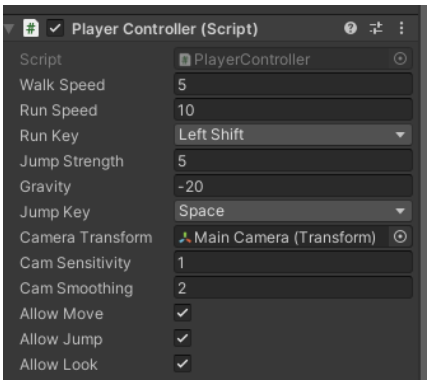

Figure 6. Screenshot of player controller

```
1 reference
void Move()
{
    currSpeed = Input.GetKey(runKey) ? runSpeed : walkSpeed;

    Vector3 forward = transform.TransformDirection(Vector3.forward);
    Vector3 right = transform.TransformDirection(Vector3.right);

    float curSpeedX = currSpeed * Input.GetAxis("Vertical");
    float curSpeedY = currSpeed * Input.GetAxis("Horizontal");
    moveDirection = (forward * curSpeedX) + (right * curSpeedY);
    moveDirection.y = yVel;

    // Move the controller
    cc.Move(moveDirection * Time.deltaTime);
}
```

Figure 7. Screenshot of code 3

The PlayerController script contains many features such as jumping, gravity, looking around in first person view, and simply moving around. When moving around, you can either walk or run activated using the run key (holding left shift as default). The code first needs to determine the “right” and “forward” of the current player. The first person camera look feature allows us to look around the scene thus changing which way is forward and which way is our right side. We use Unity’s “TransformDirection” function to convert a direction to a new player's local direction of where our player is truly looking (simply by example, looking southwest of the world now becomes our player’s forward direction since that is where we’re looking). We use this new forward and right direction to now control the player moving around. By using Input.GetAxis(“Vertical”) to get input forward and back movement using the W and S or Up and Down arrow keys, and Input.GetAxis(“Horizontal”) using the A and D or Left and Right arrow keys to move left and right. These inputs return -1 or 1 based on the key we are pressing and multiply this number by the forward and right direction which we then finally assign to the Unity character controller “Move” function to tell the player which direction we are moving. As an example, if we are looking South of the world, South now becomes our player’s forward

direction, so pressing S or Down arrow key returns a number -1 so forward direction multiplied by the negative 1 puts the player moving backwards aka the world North direction.

4. EXPERIMENT

I would like to test how my program works among the general audience, both professionals and casual users. This will be important because usability and likeability of the app for a broad range of users means it meets the needs of different skill levels and expectations. In so doing, I would be in a position to ascertain the barriers involved, integrate certain feedback into my improvement, and thus come out with an inclusive, easily accessible app, serving its purpose for all concerned.

For implementing the experiment, I have prepared a survey and gave access to the application to the participants. Participants consisted of professional designers to casual users; it can present a wide range of audiences. They were assigned tasks regarding importing 3D models, navigating the showroom, using the AI generator to create descriptions of cars, and giving feedback on usability, features, and the challenges they have encountered. This setup allowed me to see how well the application addressed both professional and amateur users. This was an upfront design to expose strengths, identify flaws that needed more improvement, and ensure the functionality and appeal were prepared for a larger audience.

Alex	Yes	The app made it super easy to import and view my 3D car designs without any hassle.
Jordan	Yes	The AI descriptions helped me come up with unique car features I hadn't thought of.
Taylor	No	I couldn't use it because my models weren't in the required glTF format.
Sam	Yes	Walking around the showroom gave me a better perspective of my designs.
Jamie	No	The interface was confusing, and I struggled to navigate the menus.
Chris	Yes	I loved the ability to customize the showroom's background for different vibes.
Morgan	Yes	The app saved me hours of brainstorming with its AI-generated ideas.
Casey	No	I found the importing process too technical for my beginner-level skills.
Drew	Yes	Scaling the models to fit perfectly in the showroom was a game changer for my presentations.
Reese	No	The AI descriptions were generic and didn't feel tailored to my needs.
Pat	Yes	It worked perfectly for my professional car modeling needs with easy file management.
Jordan	Yes	The realistic background with moving traffic made my designs feel alive.
Riley	No	The app lagged on my older computer, making it hard to use.
Quinn	Yes	I enjoyed walking around the showroom to inspect every angle of my design.
Dana	No	It was frustrating that non-glTF files couldn't be directly imported.
Cameron	Yes	The appâ€™s AI saved me time by giving me a solid starting point for new designs.
Parker	Yes	I appreciated the flexibility in setting my own controls and preferences.
Skyler	Yes	The app was simple enough for my single use case without any unnecessary features.
Jess	Yes	The showroom felt immersive, and the interactive features kept me engaged.
Blake	Yes	The app made it easy to show my designs to clients without using expensive tools.

Figure 8. Table of the experiment

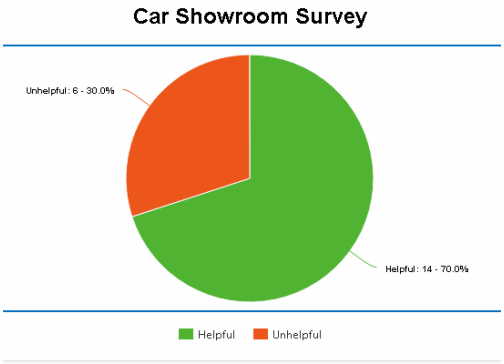

Figure 9. Car showroom survey

In the survey of 20 users, 14 users found the application helpful at 70%, while 6 did not find it helpful at 30%. This reception is fairly strong, and many liked the ease with which the application simplified model visualization and gave creative design ideas. For instance, Alex appreciated how easy it was to import a model, while Chris found the setup of showroom

backgrounds very useful. Moreover, Morgan praised the fact that AI really saved much time while getting great, creative design prompts. Finally, in 30% this was not helpful, mainly for some kind of technical problems and incompatibility-like Taylor and Dana not catching glTF as an expected file format, while Riley was pretty slow on the old hardware. It generally looks to keep the promise of being helpful for pros and end-users alike. For most of the few noted technical challenges, the strong focus on accessibility and creativity overshadows them. Certainly, addressing pain points like format flexibility or optimization for older systems would go a long way in further easing adoption.

5. RELATED WORK

This paper discusses the usability factors that influence the process of learning 3D modeling in VR and identifies such factors as interactive quality, dynamic compatibility, and flow effects [8]. The research explores these factors in the effort to enhance the learning experience in VR-based 3D modeling education.

While this research focuses on a very specific usage of 3D modeling in VR-for educational means - our project focuses on a virtual showroom for car model visualization. Both are conceived to interact in immersive environments with 3D models. Nevertheless, our application incorporates an AI content generator, which helps users build up descriptions of cars-which is a functionality not found within the VR educational tool-and with options for customized backgrounds and user-defined controls beyond that developed for educational parameters in the VR study.

The paper illustrates how AI is influencing design for creativity and efficiency, from automating repetitive tasks and generating new variations in design to giving insights into consumer preference that allows data-driven personalized design solutions [9].

Both this research and your project use AI to enhance the design process. While the research covers a wide domain of AI applications in the design industry, focusing on the aspects of automation and data-driven personalization, our application will specifically use AI for generating descriptions of cars in order to help designers conceptualize certain features. Further, our system provides a virtual showroom for immersive visualization-a component not covered by the broad view of the research.

This paper proposes some usability principles that can be applied to user interfaces of complex 3D parametric architectural design and engineering tools to be able to improve user experience, since working and navigating within complex 3D environments is challenging [10].

While the focus of this research is on architectural design tools, it does share common ground with our project in working within complex 3D environments. Both systems require intuitive user interfaces for easier interaction with models. Our application is focused on the automotive industry and will have a virtual showroom and AI-generated content to support designers. In addition, our system is targeted for both professional and casual users, while the study focuses on tools that are mainly for professional architects and engineers.

6. CONCLUSIONS

During the making of my project, I discovered some limitations which hindered the effectiveness of my app. First, I feel like I could have improved on creativity in making this app, with better usage of models and designs. I also feel like my lack of familiarity with Unity has caused some

issues in the building and designing of the app. There are also a lack of models that I could find on the unity asset store, which meant that some of the models felt out of place in the app. Another place I could have improved on is a character model to view instead of a blank model, which may enhance user experience. The main building itself feels like it lacks details compared to the other buildings on the map, which I would like to fix. Some of the other limitations are my lack of experience in making UIs and scripts. The app can also only import in GLTF files, which makes it inconvenient for certain users to upload their models [15]. Another issue is that the AI descriptions sometimes didn't feel unique or authentic, making it difficult for the user to utilize it. The AI could instead put out a variety of descriptions to make choosing one easier, instead of having the user repeatedly regenerate prompts.

REFERENCES

[1] Mountstephens, James, and Jason Teo. "Progress and challenges in generative product design: A review of systems." Computers 9.4 (2020): 80.

[2] Huang, Hsinfu, and Chang-Franw Lee. "Factors affecting usability of 3D model learning in a virtual reality environment." Interactive Learning Environments 30.5 (2022): 848-861.

[3] Adeleye, Israel Olamilekan. "The Impact of Artificial Intelligence on Design: Enhancing Creativity and Efficiency." Journal of Engineering and Applied Sciences 3.1 (2024): 1-13.

[4] Lee, Ghang, et al. "Usability principles and best practices for the user interface design of complex 3D architectural design and engineering tools." International journal of human-computer studies 68.1-2 (2010): 90-104.

[5] Fitria, Tira Nur. "Artificial intelligence (AI) in education: Using AI tools for teaching and learning process." Prosiding Seminar Nasional & Call for Paper STIE AAS. Vol. 4. No. 1. 2021.

[6] Cioffi, Raffaele, et al. "Artificial intelligence and machine learning applications in smart production: Progress, trends, and directions." Sustainability 12.2 (2020): 492.

[7] Mueller, Christoph, and Vitaliy Mezhuyev. "AI models and methods in automotive manufacturing: a systematic literature review." Recent innovations in artificial intelligence and smart applications (2022): 1-25.

[8] Kamran, Sayed Suhaib, et al. "Artificial intelligence and advanced materials in automotive industry: Potential applications and perspectives." Materials Today: Proceedings 62 (2022): 4207-4214.

[9] Luckow, Andre, et al. "Artificial intelligence and deep learning applications for automotive manufacturing." 2018 IEEE International Conference on Big Data (Big Data). IEEE, 2018.

[10] Hudson, Nathaniel, et al. "Investigating how online help and learning resources support children's use of 3D design software." Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. 2018.

[11] Remondino, Fabio. "Heritage recording and 3D modeling with photogrammetry and 3D scanning." Remote sensing 3.6 (2011): 1104-1138.

[12] Robinet, Fabrice. "glTF: Designing an open-standard runtime asset format fabricerobinet, Re mi Arnaud, Tony Parisi, and Patrick Cozzi." GPU Pro 360 Guide to 3D Engine Design. AK Peters/CRC Press, 2018. 243-260.

[13] Schilling, Arne, Jannes Bolling, and Claus Nagel. "Using glTF for streaming CityGML 3D city models." Proceedings of the 21st International Conference on Web3D Technology. 2016.

[14] Auger, Tom, and Emma Saroyan. "Overview of the OpenAI APIs." Generative AI for Web Development: Building Web Applications Powered by OpenAI APIs and Next. js. Berkeley, CA: Apress, 2024. 87-116.

[15] Bravo, Sergio Almajano. Development of a GLTF File Importer for Unreal Engine 4. Sheffield Hallam University, 2019.