SHAPEGEN TEXT TO 3D MODEL GENERATOR

Arjun T Prakash *[1], Shijo Shaji *[2], Sainath R,aghunath *[3], Arjun K S *[4], Dr.Rekha K.S *[5]

arjun$_{b}$22225*cs*@*ce* − *kgr.org, shijo*$_{b}$21160*cs*$_{b}$@*ce* − *kgr.org, sainath*$_{b}$21003*cs*$_{b}$@*ce* − *kgr.org, arjun*$_{b}$22074*cs*@*ce* − *kgr.org, rekhaks*@*ce* − *kgr.org*

College Of Engineering,Kidangoor

ABSTRACT: *This project introduces an AI-driven Text-to-3D Model Generation System that transforms natural language descriptions into fully realized 3D models using deep learning techniques. The system employs Stable Diffusion to convert text prompts into 2D images, which are then processed using depth estimation, Neural Radiance Fields (NeRF), and Signed Distance Functions (SDF) to construct an initial 3D representation. AI-driven topology optimization and texture mapping further enhance the model, ensuring realistic structure and efficient polygonal design. Users can refine shape, texture, and lighting through Python-based automation and AI-assisted sculpting tools in Blender and ZBrush. To maximize usability, the system supports exporting models in standard formats (OBJ, FBX,), making them compatible with game engines (Unity, Unreal), AR/VR platforms, animation software, and 3D printing. This approach reduces manual effort, accelerates 3D asset creation, and enhances creative flexibility. The project integrates deep learning frameworks such as PyTorch and TensorFlow, improving accuracy and realism in 3D model generation. Future enhancements include real-time interactive sculpting, advanced texture synthesis, and AI-based model refinement. this system revolutionizes 3D content creation, making it more accessible, efficient, and automated for game developers, artists, and digital designers.*

Keywords: Stable Diffusion;PyTorch;Blender;Depth Estimation;

1. INTRODUCTION

The field of 3D modeling and design has traditionally been a complex and skill-intensive process, requiring expertise in specialized software such as Blender, Maya, and 3ds Max. However, with advancements in Artificial Intelligence (AI) and Machine Learning (ML), new techniques have emerged that allow for the generation of 3D models from textual descriptions[1]. This transformation is made possible through AI models like Stable Diffusion, GANs (Generative Adversarial Networks), and deep neural networks that can interpret text and create corresponding visual representations[2][3]. To address the challenges of traditional 3D modeling Shapegen,introduces a text-to-3D workflow that converts natural language prompts into detailed 3D models. By utilizing Stable Diffusion models for image generation and Python-based tools for model refinement, Shapegen simplifies the 3D creation process and makes it accessible to designers, developers, and researchers[4].

2. LITERATURE SURVEY

The Shapegen project is a comprehensive body of research spanning natural language processing (NLP), computer vision, 3D modeling, and generative AI, which collectively enable the transformation of text prompts into detailed 3D models[9].The core of this project is the integration of text-to-image generation technologies, particularly Stable Diffusion and a latent diffusion model that produces high-quality images from textual descriptions[14].

Stable Diffusions ability to generate detailed and realistic images while maintaining computational efficiency makes it a foundation for Shapegen text-to-3D pipeline[19]. Similarly, DALL-E and CLIP (Contrastive Language–Image Pretraining) have demonstrated the potential of combining NLP with computer vision, learning the relationship between text and visual data to generate images that align with textual prompts[7]. These advancements inspire Shapegen's approach to converting natural language descriptions into visual representations, which serve as the foundation for 3D model construction.

A comparative evaluation of unsupervised anomaly detection algorithms highlights the need for tailored detection methods for various anomaly types. Key algorithms discussed include k-NN, Local Outlier Factor (LOF), and One-Class Support Vector Machines (SVM)[11]. The findings suggest that while histogram-based methods excel in real-time applications, nearest neighbour algorithms generally outperform clustering methods, emphasizing the importance of selecting the right algorithm based on dataset characteristics[3].

Also the users can easily customize the generated 3D models to meet specific requirements, such as adjusting the shape, size, and texture. As AI continues to evolve, we can expect even more sophisticated and powerful 3D modeling tools to emerge, further streamlining the process and opening up new creative possibilities[7]. research explores a novel approach to 3D model generation that leverages the power of Stable Diffusion, a state-of-theart text-to-image model. By inputting a textual description of a desired 3D object, the model generates a sequence of 2D images that capture the object from multiple perspectives[5]. These 2D images are then processed and transformed into a 3D point cloud, a fundamental representation of 3D geometry[1].

This innovative approach, powered by AI, offers a significant advancement in 3D modeling, making it accessible to a wider audience and streamlining the process[7]. By leveraging the capabilities of Stable Diffusion, we can bypass traditional 3D modeling techniques that require specialized software and extensive technical skills. This AI-driven approach empowers users to create complex and realistic 3D models directly from text prompts, opening up new possibilities for creativity and innovation in various fields including game development, film production[6].

The finetuning step allows for greater customization and control over the final 3D model, resulting in highly detailed and realistic 3D representations[8]. This iterative process of training and refinement ensures that the generated 3D models are accurate, consistent with the original text prompt, and visually appealing[10]. By incorporating advanced machine learning techniques, we can push the boundaries of 3D model generation and create increasingly sophisticated and lifelike digital objects.

The resulting 3D model can be exported in various industrystandard formats, such as OBJ, PLY, or STL[12][15]. This enables seamless integration of the generated model into a wide range of applications, including game development, virtual reality, augmented reality, 3D printing[16].

The practical applications of Shapegen are supported by research in 3D modeling for gaming, animation, architecture, and product design. Studies have also shown that AI-generated 3D models can significantly reduce production time and costs in these industries by automating asset creation and enabling rapid prototyping[18]. Shapegen ability to generate 3D models from text prompts, combined with its customization and export features, makes it a valuable tool for these applications. Moreover, the project addresses key challenges in traditional 3D modeling, such as high hardware requirements and limited accessibility, by optimizing for mid-range hardware and reducing technical barriers[20].

In the significant advancements in text-to-image generation, 3D reconstruction, and AI-driven customization that form the foundation for the Shapegen project[13]. By integrating these technologies, Shapegen simplifies 3D modeling, making it more accessible and efficient while enabling rapid prototyping and iterative design.Also future can include advancements such as integrating advanced AI models, improving hardware optimization, and expanding support for industry-specific applications[17].

3. METHODOLOGY

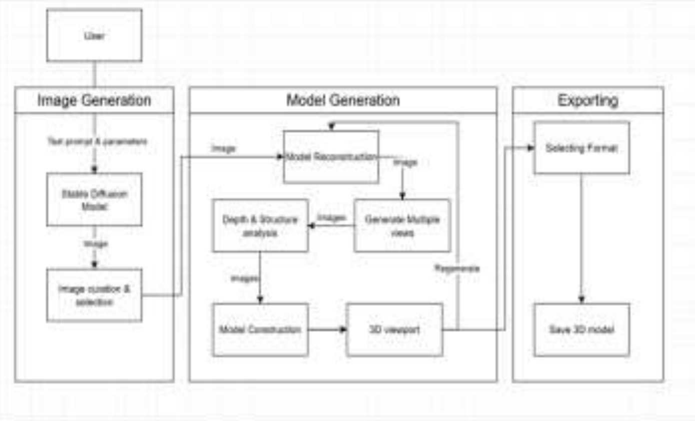

Figure 1: Architecture

1. problem Definition: The primary objective of this research is to develop a Text-to-3D Model generation system using Stable Diffusion and image-to-3D conversion techniques. Traditional 3D modeling requires specialized software and skills, making it inaccessible to many users. This project aims to lower these barriers by allowing users to input textual descriptions and generate 3D models automatically. The process involves converting text prompts into 2D images and then using image-based 3D reconstruction to create a structured 3D model.

2. Data Collection: Data collection is a crucial step in the Text-to-3D Model project, as it ensures high-quality inputs for generating accurate 3D models. The data includes text prompts, 2D images, and 3D models, which are used for training and validation.

Text Prompts Collection: Text descriptions are gathered from datasets like LAION-5B, OpenAI CLIP, and COCO, covering diverse objects (e.g., vehicles, furniture, animals).Manually created prompts ensure complex and creative object descriptions for training.

2D Image Collection: Stable Diffusion generates multiple 2D images of objects from text.Real-world images from datasets like ImageNet and Google Scanned Objects improve model accuracy.Multi-view images are created to enhance 3D reconstruction.

3D Model Dataset: Existing 3D models from sources like ShapeNet, Objaverse, and Google Scanned Objects serve as reference data.The models are preprocessed into standard formats (.OBJ, .STL, .GLB) and cleaned for consistency.

Data Labeling and Annotation: Text-to-3D mappings ensure correct alignment between descriptions and generated models.Multi-view annotations help in reconstructing accurate 3D structures.

3. Data Preprocessing: Data preprocessing is essential in the Text-to-3D Model project to ensure that the input text, images, and 3D models are clean, structured, and suitable for model training. This phase involves text processing, image preprocessing, and 3D model standardization to improve accuracy and efficiency.

Text Processing: Since the system converts text prompts into 3D models, the textual descriptions need to be cleaned, structured, and embedded for AI processing.
Steps in Text Preprocessing:

- Text Cleaning
- Tokenization and Embedding
- Attribute Extraction

Image Preprocessing: Since Stable Diffusion generates 2D images before converting them into 3D models, the generated images must be cleaned and enhanced.

Steps in Image Preprocessing:

- Image Filtering and Selection
- Super-Resolution Enhancement
- Multi-View Consistency

3D Model Standardization: The 3D models collected for training must be converted into a uniform format to ensure consistency.

Steps in 3D Model Standardization:

- File Format Standardization
- Mesh Cleaning and Optimization
- Texture Mapping and UV Unwrapping
- Normalization and Scaling

Data Augmentation: To improve model robustness, data augmentation techniques are applied.Generating paraphrased descriptions to improve the AI's ability to understand different ways of describing the same object.Creating variations of images by adjusting brightness, contrast, and rotation.Slightly modifying shapes, textures, and scaling to increase training diversity.

4. Model Architecture: The Text-to-3D Model system consists of two main components:

- Text-to-Image Generation
- Image-to-3D Reconstruction

Text-to-Image Generation: The first step is generating 2D images from text descriptions using Stable Diffusion or similar models.Components Natural Language Processing (NLP) Model,CLIP (Contrastive Language-Image Pretraining)converts the text input into a latent space representation.This allows to understand the meaning of the prompt in terms of visual features.Stable Diffusion Model (Text-to-Image Model) Uses the CLIP embedding to generate a realistic 2D image based on the text prompt.Multiple viewpoints of the object (front, side, top) are generated.Multi-View ControlNet (for Multi-Angle Images)Ensures the different angles of the same object are consistent.

Image-to-3D Reconstruction: Once the 2D images are generated, they are converted into a 3D model using deep learning techniques.Depth Estimation Model (Single Image to 3D).A Neural Network predicts a depth map from the generated image.This depth map is then converted into a 3D point cloud.Models like MiDaS or DPT (Depth Prediction Transformers) are used.Multi-View Stereo (MVS) for 3D Shape Reconstruction.If multiple views of an object are available, MVS aligns them to form a complete 3D structure.Ensures that all viewpoints merge into a realistic 3D object.Neural Radiance Fields (NeRF) for Volume-Based 3D Generation.NeRF generates a volumetric 3D representation from multiple images.It estimates the color and density at different points in 3D space, allowing for smooth 3D shape formation.Mesh Generation (Point Cloud to 3D Model).Once the 3D point cloud is created, it is converted into a polygonal 3D model.Marching Cubes Algorithm – Converts the point cloud into a surface mesh.Poisson Surface Reconstruction – Improves smoothness and accuracy.Texture Mapping and UV Unwrapping a GAN-based texture model is used to apply textures to the 3D surface.UV unwrapping ensures the textures align properly on the 3D model.

5. Model Training And Fine-Tuning: Model training and fine-tuning are critical steps in the Text-to-3D Model project. This phase focuses on training AI models to generate high-quality 2D images from text and then converting these images into accurate 3D models. The training process consists of pretraining on large datasets, fine-tuning on custom datasets, and applying optimization techniques for better accuracy and efficiency.

Pretraining on Large Datasets: To build a strong foundation, the model is pretrained on large-scale datasets that contain text-image pairs and 3D models.

- Text-to-Image Pretraining

LAION-5B (5 billion text-image pairs for training Stable Diffusion).MS COCO and OpenAI's CLIP dataset for text-to-image generation.Pix3D for fine-grained text-to-2D image alignment.2D images generated during training are used for 3D reconstruction in the next phase.

Fine-Tuning with Custom Data: Once pretrained, the model is fine-tuned on domain-specific datasets to improve accuracy for specific applications (e.g., medical, gaming, architecture).

- Custom Dataset Collection
- Supervised Fine-Tuning
- Transfer Learning

Collect text descriptions, AI-generated 2D images, and corresponding 3D models.Focus on specific categories like vehicles, furniture, animals, architectural structures, or fantasy objects.
Adjust model parameters based on the collected dataset to enhance its ability to generate more realistic and detailed 3D models.
Implement CLIP-based fine-tuning to enhance the understanding of complex text prompts.Fine-tune the NeRF model or 3D reconstruction model to improve multi-view consistency.

Optimization Technique:To improve accuracy, quality, and speed, several optimization techniques are applied.

- Contrastive Learning
- Adversarial Training (GANs)
- Multi-View Consistency Loss

CLIP uses a contrastive loss function to align text descriptions with their corresponding images in a shared embedding space.Helps the model differentiate between similar objects (e.g., distinguishing between "a cat statue" and "a real cat").
Uses a Generative Adversarial Network (GAN) to refine textures and increase realism.GANs help remove noise and improve textures in the generated 3D model.
Ensures that all 2D views of an object align correctly in 3D space.Reduces issues like misalignment, blurriness, and texture inconsistencies.

Optimization for Performance: Since Stable Diffusion and 3D reconstruction models require high computational power, optimization techniques are used to improve efficiency.

- Pruning and Quantization

Removes unnecessary model parameters to reduce complexity.Compresses the model to reduce memory usage.

6. Model Evaluation: Model evaluation is a crucial step in the Text-to-3D Model project to ensure the generated 3D models are accurate, realistic, and aligned with the input text descriptions. The evaluation process involves quantitative metrics, qualitative analysis, and user testing to assess the model's performance.

Quantitative Evaluation Metrics: These metrics measure how well the generated 3D model matches reference models and maintains structural integrity.

- Chamfer Distance (CD)
- Structural Similarity Index (SSI)
- Intersection over Union (IoU)
- Fidelity Score (FS)
- Multi-View Consistency Score

Measures how closely the generated 3D model's point cloud matches a reference 3D model.Lower values indicate better accuracy in reconstructing the object.
Evaluates the shape and structural consistency of the generated 3D model compared to the expected reference model.SSI greater than 0.8 indicates a strong similarity.
Measures the overlap between the generated 3D model and a ground-truth 3D model.
Measures how well the 3D model captures the key features described in the text prompt.Higher scores indicate that the model generates accurate and visually appealing 3D outputs.
Evaluates whether different 2D views (top, side, front) align correctly when reconstructed into 3D.Ensures the generated object looks realistic from all angles.

Qualitative Evaluation: Since 3D models need to be visually accurate, qualitative evaluation is necessary to assess.

- Visual Comparison with Reference Models
- Texture and Material Realism
- User Perception and Aesthetic Quality

The generated 3D model is compared to real-world 3D models (from datasets like ShapeNet).Experts review the shape, texture, and proportions.
Checks if the generated textures align properly with object surfaces.Assesses whether the color, material, and lighting are consistent.
Evaluates how visually appealing and realistic the 3D model appears to human observers.Conducted through user surveys and expert feedback.

User Study and Feedback Collection: A key evaluation step involves testing how well users can interpret and use the generated 3D models.

- User Testing
- Real-World Application Testing

Users describe objects and rate the accuracy of generated models.Key evaluation factors,Accuracy (Does it match the description?),Detail Level (Does it have the right textures?),Usefulness (Is it usable for gaming, VR, or 3D printing?).
The 3D models are tested in game engines (Unity, Unreal Engine), AR/VR environments, and 3D printing software.Checks if the models integrate correctly without errors.

Continuous Model Improvement: If models fail to match text prompts, the dataset is fine-tuned with additional training.Models are continuously retrained on failed cases to improve output quality.

7. Implementation and Integration: The implementation of the Text-to-3D Model system is designed as a Python-based end-to-end pipeline, integrating text-to-image and image-to-3D conversion models into a seamless workflow. The system includes a Python-based frontend, a backend API, and a 3D rendering module.

System Architecture Overview: The architecture consists of three main components.

- Frontend
- Backend
- 3D Rendering and Export

A web interface for text input and 3D visualization.Manages text processing, image generation, and 3D model creation.Allows users to view, modify, and download the generated 3D model.

Frontend Implementation: The frontend is built using Python, providing an interactive interface for users to input text descriptions and preview the generated 3D model.

Backend Implementation: The backend is developed using Python and handles text-to-image processing, 3D reconstruction, and model storage.
Backend Tech Stack:.Stable Diffusion (Text-to-Image Generation) – Converts text input into multiple 2D images.Neural Radiance Fields (NeRF) / Depth Estimation – Converts images into a 3D point cloud and mesh model.Pytorch and TensorFlow – Used for training AI models.Open3D – Processes 3D point clouds and meshes for final model refinement.
Backend Workflow: Receive text input from the frontend API.Process text with CLIP model to generate a text embedding vector.Use Stable Diffusion to generate multiple 2D images from the text.Run depth estimation (MiDaS) or NeRF to create a 3D point cloud.Mesh generation and texture mapping.Convert point cloud to 3D mesh using Marching Cubes or Poisson Surface Reconstruction.Apply textures using GAN-based models for realism.Save and return the 3D model in OBJ, GLB, or STL formats.

3D Visualization and Export: Once the 3D model is generated, it needs to be visualized and exported for real-world applications.Users can rotate, zoom, and inspect the 3D object.The final 3D models can be exported in different formats .OBJ and .STL (For 3D printing and CAD applications),.GLB / .gltf (For VR/AR applications and game engines like Unity and Unreal).

4. RESULTS AND DISCUSSION

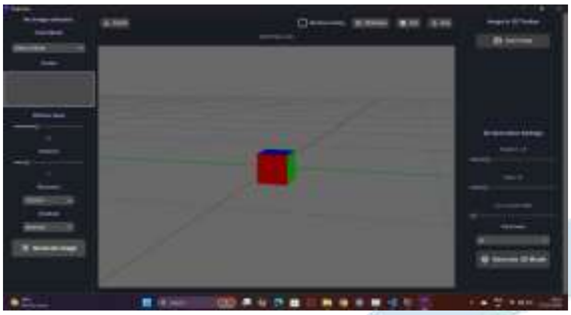

Figure 2:Default view.

The implementation of the Text-to-3D Model generation tool with an intuitive user interface. It includes multiple adjustable parameters to control both the image generation and 3D model conversion processes. Let's go through each section of the interface in detail.

1. Left Panel – Image Generation Controls: This section is primarily used for generating images from a text prompt using a diffusion-based AI model.

Select Model (Dropdown): This dropdown allows users to choose a pre-trained AI model for image generation.Different models may provide different styles, levels of detail, or functionalities.Example models could include Stable Diffusion, Deep Dream, or custom-trained models.
Prompt (Text Input): This is where users type a description of the image they want to generate.The AI model interprets the text and tries to generate an image that matches it
Diffusion Steps (Slider): Controls how many steps the AI takes to refine the image.Higher values produce more detailed and accurate images but take longer.Lower values generate images quickly but with lower quality and more noise.Default setting: 25 steps (as shown in the UI).
Guidance (Slider): This parameter determines how strictly the model follows the given prompt.Lower values (e.g., 2-3) allow more creative freedom but might not match the prompt exactly.Higher values (e.g., 7-10) force the image to match the prompt closely but can make it look unnatural.Default setting in your UI: 3 (which is moderate guidance).
Resolution (Dropdown): Users can select the resolution of the generated image.Higher resolutions (e.g., 1024x1024) produce more detailed images but take longer to generate.Your UI currently shows 512x512, which is a good balance between quality and speed.
Scheduler (Dropdown): Determines how the AI model removes noise in the diffusion process.The ”Balanced” option is selected in your UI, which likely means a compromise between speed and quality.Other possible options could include:Fast (for quicker but rougher images).High-Quality (for more refined images with extra processing time)
Generate Image (Button): Clicking this button starts the image generation process based on the provided prompt and settings.Once the image is created, it can be used for 3D model conversion.

2. Right Panel – 3D Model Generation Control: This section focuses on converting an image into a 3D model, allowing users to fine-tune parameters for the best 3D output.
Import Image (Button): Allows users to upload a custom image instead of generating one from text.This is useful if the user has an external image they want to convert into a 3D model.
3D Generation Settings: This section contains parameters for fine-tuning how the 3D model is generated.

Guidance (Slider): Similar to the guidance in the image generation panel, this controls how closely the 3D model follows the input image.A higher value ensures the model is very similar to the image but may introduce artifacts.A lower value allows more flexibility but may not fully resemble the original image.The default value in your UI is 4.8, which is a moderate setting.
Steps (Slider): Determines the number of iterations used to generate the 3D model.More steps improve quality but increase processing time.The UI currently shows 25 steps, similar to the image generation process.
Face Counts (Slider): This parameter controls how detailed the 3D model is by setting the number of faces (polygons) in the mesh.Lower face count results in a simpler, lower-detail model (better for performance).Higher face count creates a smoother and more detailed 3D model.The UI shows 6000 faces, which is a reasonable balance for real-time applications.
File Format (Dropdown): Determines the format in which the 3D model will be saved.The default format in your UI is .glb (GLTF Binary), which is widely used in 3D applications, game engines, and web-based 3D rendering.Other formats might include:.obj (common format, widely supported),.fbx (used in professional 3D software like Blender and Maya)
Generate 3D Model (Button): Clicking this button starts the 3D model creation process based on the selected image and settings.The output can be downloaded and used in applications like Blender, Unity, Unreal Engine, or WebGL.

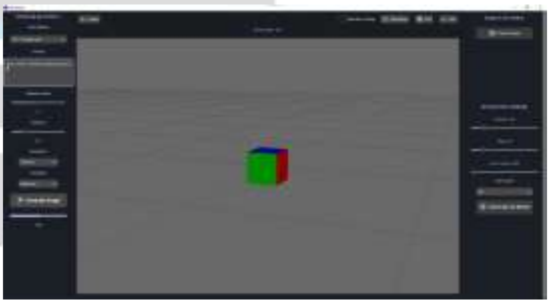

Figure 3:Progress bar updation.

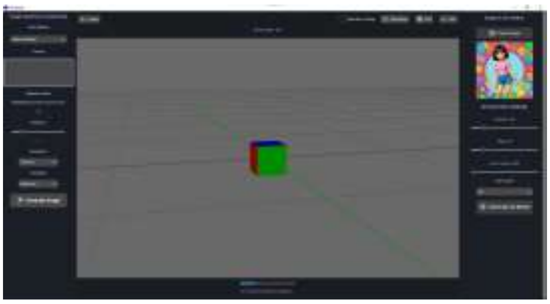

Figure 4:Image generation according to text prompt.

It is a image generated from our project using Stable Diffusion.Stable Diffusion is a deep-learning-based generative AI model that transforms text prompts into high-quality images. Based on the UI and the image selection window we provided, our project appears to use Stable Diffusion for text-to-image generation before converting the image into a 3D model.

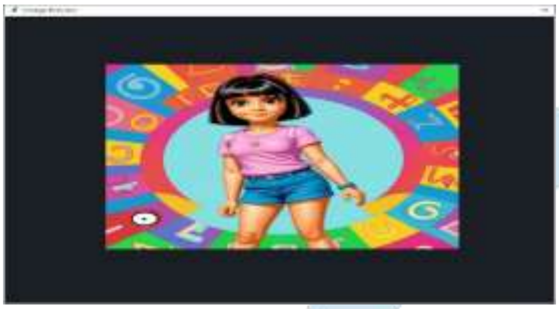

Figure 5:Preview.

Figure 6:Loading Stable Diffusion.

Figure 7:Loading HUNYUAN 3D.

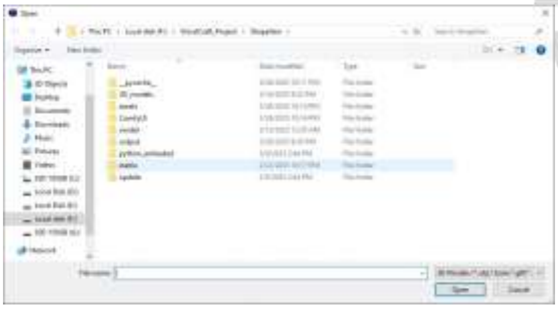

Figure 8:Import 3D-Model.

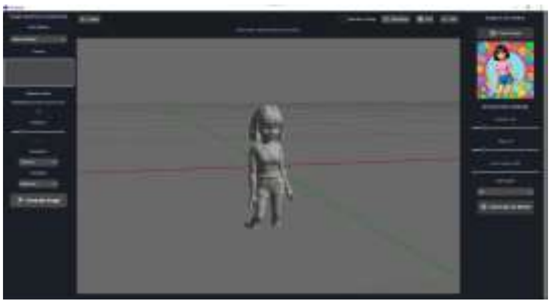

Figure 9:Final Output.

As we can see in the 3D view port of our project, It generate a 3D model from the generated image after pressing Generate 3D Model Button.

5. CONCLUSION

In conclusion, we explored the state-of-the-art advancements in text-to-3D shape generation and GUI code generation. We have discussed various techniques, challenges, and future directions in these fields. Text-to-3D shape generation has made significant strides, with models like Dream3D demonstrating impressive results in generating high-quality 3D shapes from text descriptions. However, challenges remain in terms of generating complex shapes, handling diverse text prompts, and ensuring consistency between the text and the generated 3D model. The integration of multi-view stereo techniques and learning-based 3D reconstruction methods ensures the generation of high-quality, customizable 3D assets, while advancements in AI-driven optimization and program synthesis provide the tools necessary for fine-tuning and personalization.Alsot the project ability to export models in standard formats such OBJ, FBX ensures seamless integration into existing workflows, making it a valuable addition to the toolkit of designers and developer.Also future enhancements such as integrating advanced AI models, expanding support for industry-specific applications, and improving hardware optimization will further enhance Shapegen capabilities,thereby making it an cutting-edge solution for 3D design.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 126–135, 2017.

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[4] Lin Gao, Yu-Kun Lai, Jie Yang, Ling-Xiao Zhang, Shihong Xia, and Leif Kobbelt. Sparse data driven mesh deformation. *IEEE transactions on visualization and computer graphics*, 27(3):2085–2100, 2019.

[5] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[6] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander

Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International journal of computer vision*, 128(7):1956–1981, 2020.

[7] Seongmin Lee, Benjamin Hoover, Hendrik Strobelt, Zijie J Wang, ShengYun Peng, Austin Wright, Kevin Li, Haekyu Park, Haoyang Yang, and Duen Horng Polo Chau. Diffusion explainer: Visual explanation for text-to-image stable diffusion. In *2024 IEEE Visualization and Visual Analytics (VIS)*, pages 96–100. IEEE, 2024.

[8] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*, pages 740–755. Springer, 2014.

[9] David R Martin, Charless C Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE transactions on pattern analysis and machine intelligence*, 26(5):530–549, 2004.

[10] Lars Mescheder. On the convergence properties of gan training. *arXiv preprint arXiv:1801.04406*, 1(16):2, 2018.

[11] Andrew P Norton and Yanjun Qi. Adversarial-playground: A visualization suite showing how adversarial examples fool deep learning. In *2017 IEEE symposium on visualization for cyber security (VizSec)*, pages 1–4. IEEE, 2017.

[12] Anton Obukhov, Maximilian Seitzer, Po-Wei Wu, Semen Zhydenko, Jonathan Kyl, and Elvis Yu-Jing Lin. High-fidelity performance metrics for generative models in pytorch. *Version: 0.3. 0*, 13, 2020.

[13] Min Pang, Ligang He, Fengguang Xiong, Xiaowen Yang, Zhiying He, and Xie Han. Developing an image-based 3d model editing method. *IEEE Access*, 8:167950–167964, 2020.

[14] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[15] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[16] Patrick Tinsley, Adam Czajka, and Patrick Flynn. This face does not exist... but it might be yours! identity leakage in generative models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1320–1328, 2021.

[17] Xiaowen Yang, Xie Han, Qingde Li, Ligang He, Min Pang, and Caiqin Jia. Developing a semantic-driven hybrid segmentation method for point clouds of 3d shapes. *IEEE Access*, 8:40861–40880, 2020.

[18] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021.

[19] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3836–3847, 2023.

[20] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.