Text-driven generation of 3D animation using code

Harsh Sharma
Boston University
hsharma@bu.edu

Mani Mukesh Reddy Nukal
Boston University
nmani@bu.edu

Thiru Satya Surya Mahaveer Bonagiri
Boston University
mahaveer@bu.edu

Abstract

Graphic modeling remains a grueling task in the field of 3D animations. Having a program that understands natural language to deduce the expected action by the designer would pave the way to removing the skill barriers required to enter the field. To address this problem we propose to create a method which will be able to generate animations from any natural text describing an action through code.

1 Introduction

In the world of 3D animation, animators spend a laborious amount of time designing characters and minute details. Having an automated way of Generating 3D Animations would lower the barrier to entering the field and increase the efficiency of those experienced in the field. Creating Models which will be able to render code directly from text to 3d animation would help achieve a deeper understanding between natural language context and their meaning in the real world. There also exists Graphic animation software like Unreal Engine[3], in which we can use code to render objects to animate them. With the onset of large language models like the Codex[1], which can directly generate code, we can expedite the process of generating animations. This way, animators will be able to use natural language to generate 3D animations. This would increase the efficiency of 3D animation as the animator would no longer need to manually code each part of the animation they want to create. Instead, they could give the language model a context and the model would generate the code to create the desired animation. This will also help to bridge the gap between natural language and the representation of the physical world. By having a model that is able to generate code from natural language, animators will be able to create animations that accurately represent the context given. This could be used to animate characters, objects and scenes in a much more efficient and accurate way.

In this project, we try to achieve this on CARLA[2], which is an open-source simulator for autonomous driving research. We use a language model to generate code from natural language context, which renders and animates objects in the simulation.

In conclusion, the use of automated 3D animation generation will help to reduce the amount of time and effort required to create 3D animations. This will help animators to create more accurate and efficient animations in a much shorter amount of time. It will also help bridge the gap between natural language and the physical world, as the language model will be able to generate code from natural language context. This could be used to create animations of characters, objects, and scenes in a much more efficient and accurate way.

2 Related Work

Previous work in the field of text-to-3D animation has focused on primarily human centered animations. For example, the work by Hong et. al.[4][6] takes CLIP images generated using text and trains their model to generate 3D animations, but it is only able to generate human-like subjects in their animations. Similarly work by Zhang et. al.[10], a text driven human motion generation with diffusion model which demonstrates several desired properties over existing models. Previous work in the field of text to Code[8],[5] are created to solve the problem to generate code from text, but is not trained specifically to solve problems related to computer graphics.

None of the works mentioned above have been applied to the field of graphic generation. This project is the first to apply text-to-code to the field of 3D animation generation. We use a language model to generate code from natural language context, which renders and animates objects in the simulation.

The results of this project demonstrate that it is possible to generate 3D animations from natural language context. We were able to generate animations of cars and other objects in the CARLA simulator. The results of this project show that it is possible to generate 3D animations from natural language context, which can be used to create more accurate and efficient animations. This could be used to animate characters, objects and scenes in a much more efficient and accurate way.

3 Methodology

We have developed two modes for producing code given a prompt. The generation mode is used to create a code from scratch and The edit mode is useful when the user wants to modify the existing code to make it more efficient or to add new features. The user can provide the instruction to the Codex api and it will generate the modified code. The generated/modified code is then run locally to check for errors. If errors are present, the code is regenerated using the traceback error as prompt. This process is repeated until the code is error-free.

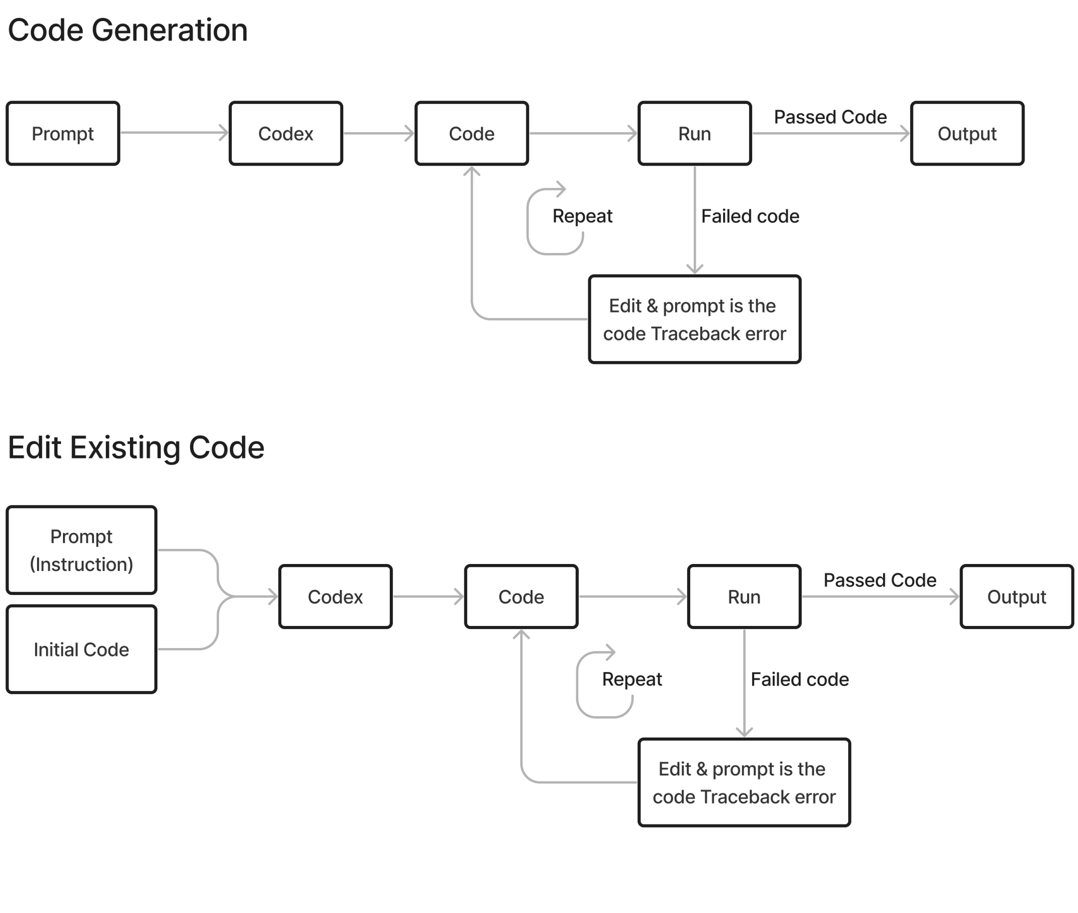

Figure 1: Flow diagram of methodology implemented

4 Results

4.1 Identifying limitations and strength of Codex

We first wanted to understand limitations of current state-of-the-art image generation model when compared to code generation using Codex. We came up with 10 adversarial examples which are performed poorly by DALL.E[7] but Codex is able to generate accurately in Opensource framework for python OpenGl[9](Fig:2).

4.1.1 Observations from this exercise

1. Codex fails to generate the desired model or creates a default mode of cube if the prompt is not in detail or complex
2. Sometimes Codex generates incomplete/incorrect code in the first iteration, reiterating again for couple of times gives the desired output
3. Codex started generating better models as we give more detail in the prompt.
4. Codex runs into infinite loop for some prompts

4.2 Rendering using prompts on CARLA

We were able to render and modify properties of objects in CARLA. As an example we were able to change color of all cars in CARLA to blue (fig:3).

Github repo: `https://github.com/97harsh/graphic_codex`

5 Limitations

1. Codex
   (a) The code generated by the Codex may be inaccurate and not able to completely do what the prompt expects it to do.
   (b) The code may undo or comment out the modifications it made in order to rectify any errors.
   (c) Codex attempts to render objects that are not available in CARLA.
2. CARLA
   (a) The current version of CARLA has inconsistencies in the attributes for similar objects.

6 Code References

Codex API Documentation: `https://beta.openai.com/docs/guides/code/introduction`
Code and model weights available for AvatarCLIP: `https://github.com/hongfz16/AvatarCLIP`
CLIP: `https://huggingface.co/docs/transformers/model_doc/clip`
Stable Diffusion: `https://huggingface.co/spaces/stabilityai/stable-diffusion`
Gpt-code-clippy : `https://github.com/codedotal/gpt-code-clippy`
StructCoder: `https://github.com/reddy-lab-code-research/structcoder`

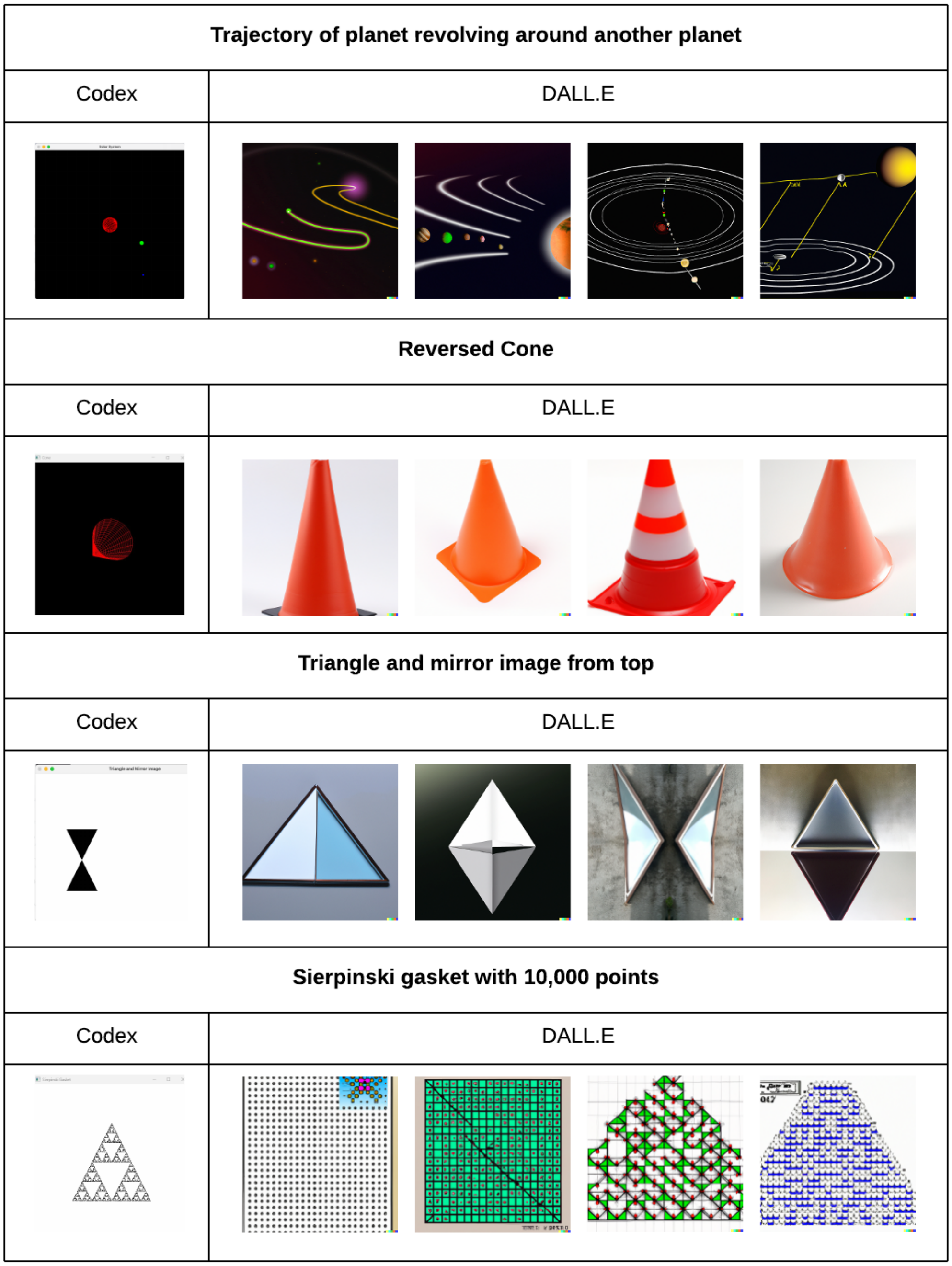

Figure 2: The following figure shows the images rendered with code generated by Codex in OpenGL and the example images that are produced by Dall.E with the same prompt.

References

[1] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[2] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on*

Figure 3: The following figure shows different examples of changes we can do with Codex in the edit mode, giving input as text prompts

Robot Learning, pages 1–16, 2017.

[3] Epic Games. Unreal engine.

[4] Fangzhou Hong, Mingyuan Zhang, Liang Pan, Zhongang Cai, Lei Yang, and Ziwei Liu. Avatarclip: Zero-shot text-driven generation and animation of 3d avatars. *arXiv preprint arXiv:2205.08535*, 2022.

[5] Shuai Lu, Daya Guo, Shuo Ren, Junjie Huang, Alexey Svyatkovskiy, Ambrosio Blanco, Colin Clement, Dawn Drain, Daxin Jiang, Duyu Tang, et al. Codexglue: A machine learning benchmark dataset for code understanding and generation. *arXiv preprint arXiv:2102.04664*, 2021.

[6] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

[7] Dhariwal P. Nichol A. Chu C. Chen M. Ramesh, A. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv.2204.06125*, 2022.

[8] Sindhu Tipirneni, Ming Zhu, and Chandan K Reddy. Structcoder: Structure-aware transformer for code generation. *arXiv preprint arXiv:2206.05239*, 2022.

[9] Mason Woo, Jackie Neider, Tom Davis, and Dave Shreiner. *OpenGL programming guide: the official guide to learning OpenGL, version 1.2*. Addison-Wesley Longman Publishing Co., Inc., 1999.

[10] Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *arXiv preprint arXiv:2208.15001*, 2022.