



"AniGEN"

**Natural Language Driven
AI Avatar Motion
with Video-Based Motion Data**

Presented by Marzukh Akib Asjad,
Mikael Ho Yu Lau & James Olano

Agenda

- 01 Background
- 02 Objectives
- 03 Methodology
- 04 Final Results
- 05 Conclusion



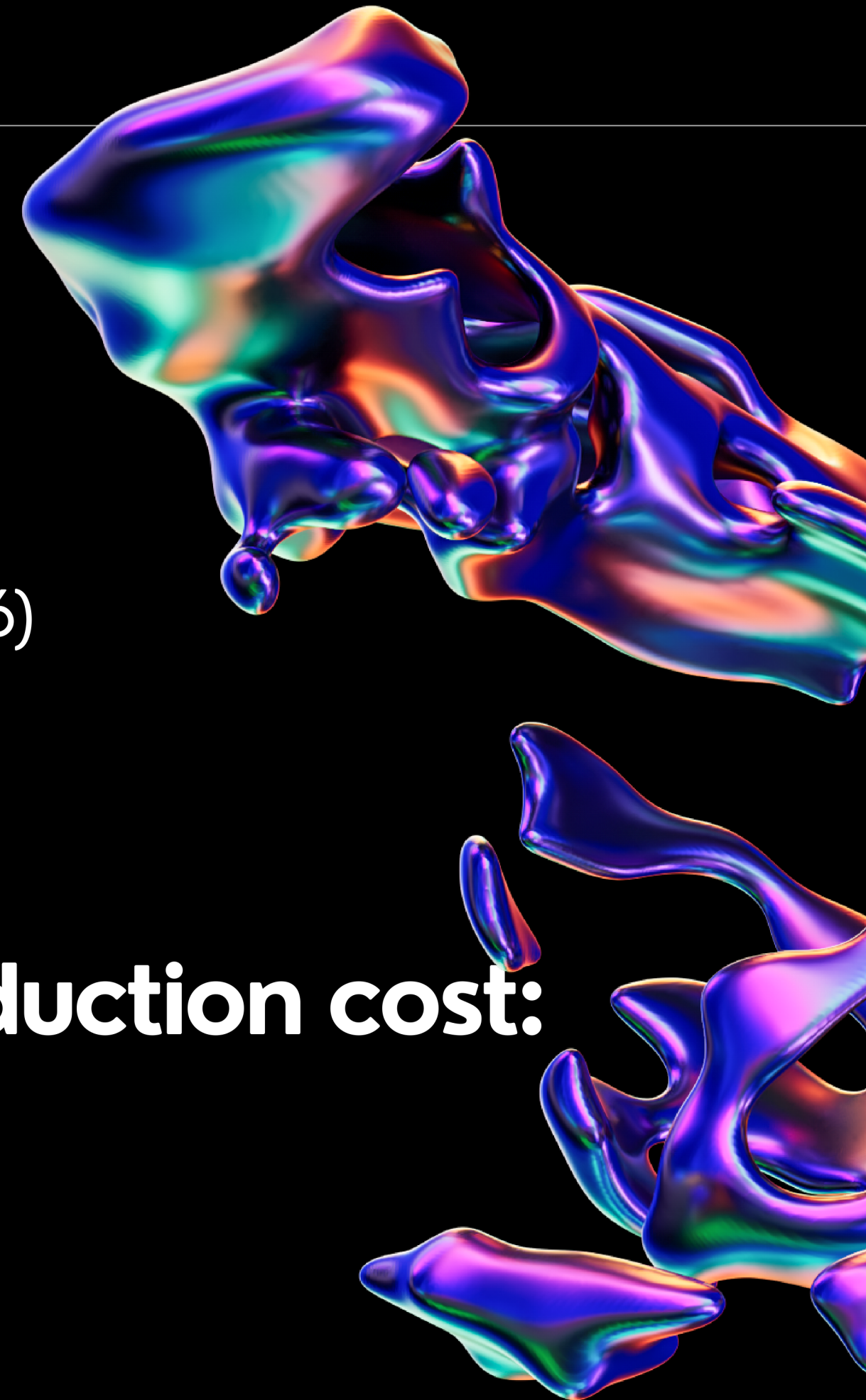
Background



Grand Theft Auto VI (GTA 6)
by Rockstar Studio

[]

Estimated production cost:
USD 6 Billion



Background

[2] Film production cost



Pirates of the Caribbean: On Stranger Tides

USD 378.5 million

Avengers: Age of Ultron

USD 365 million



Background

[3] Motion capturing technology



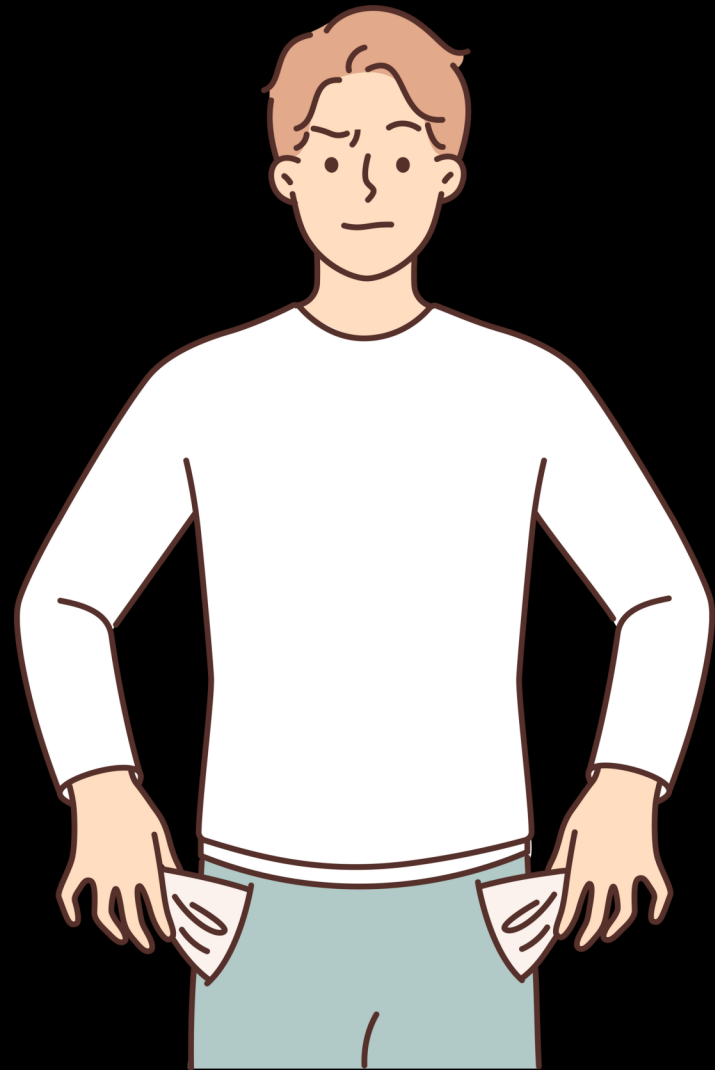
Big Companies

Advanced
equipment &
technical
knowledge
required

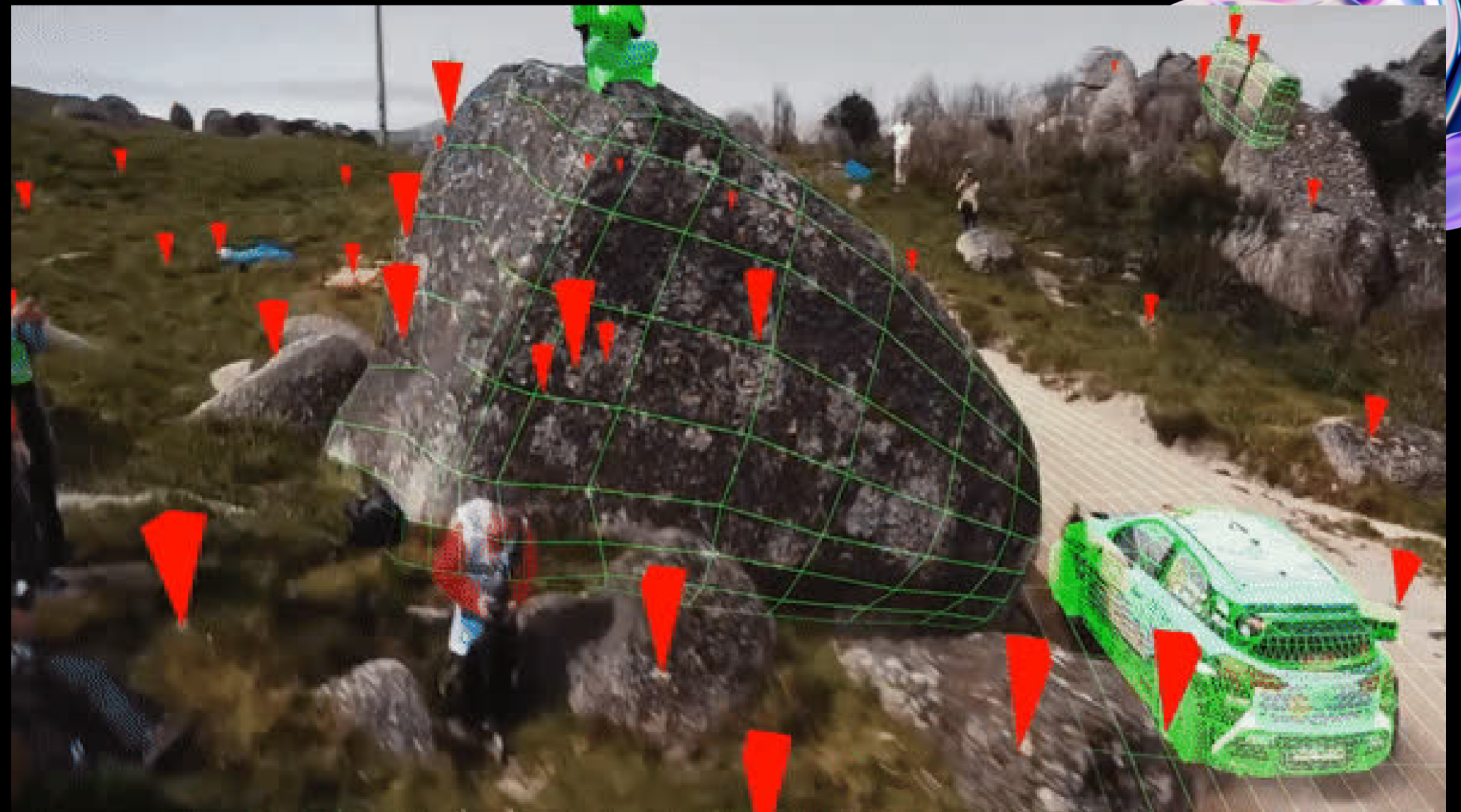


Background

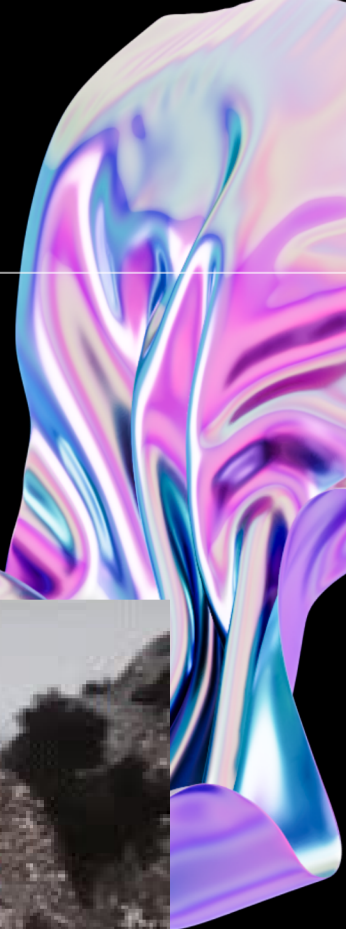
[4] Rotomation



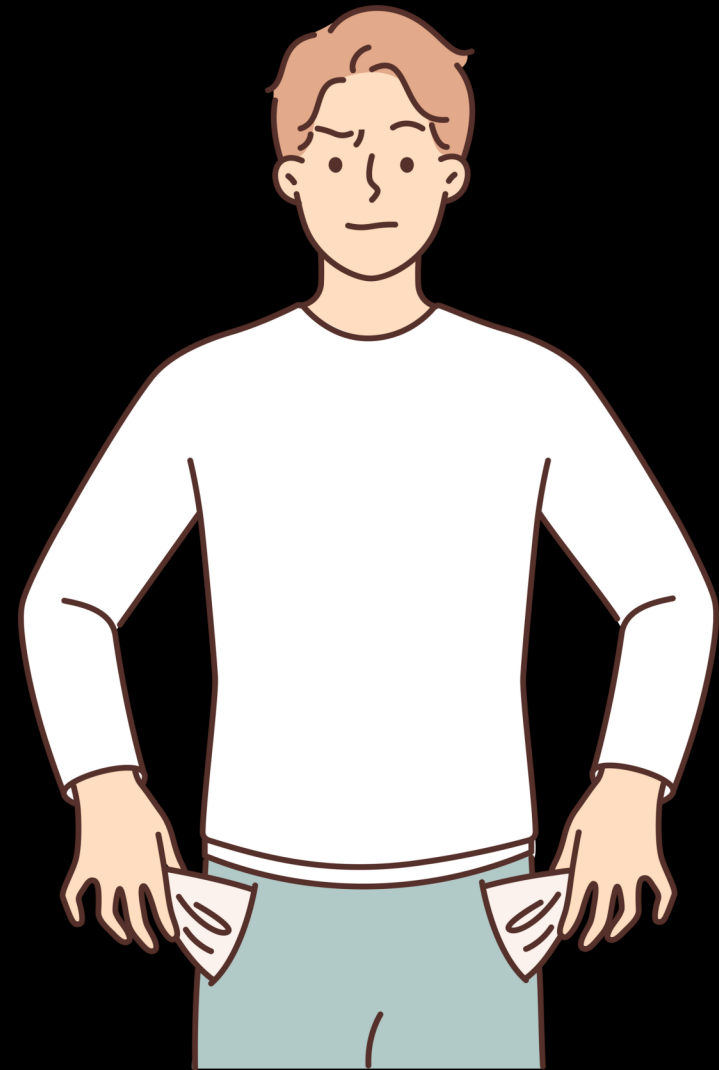
Independent Artists



Issues: Time-consuming



Background



Independent Artists

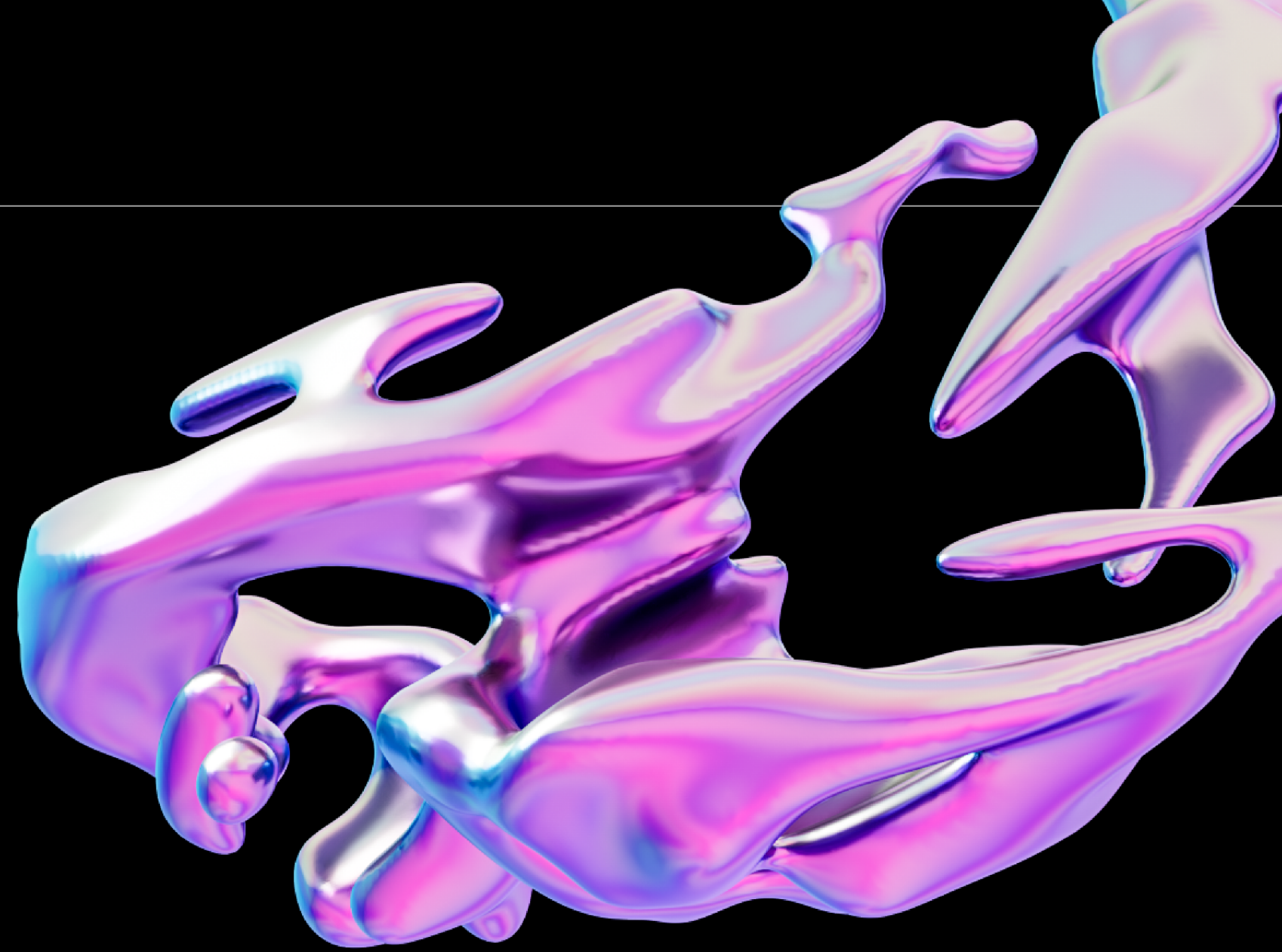
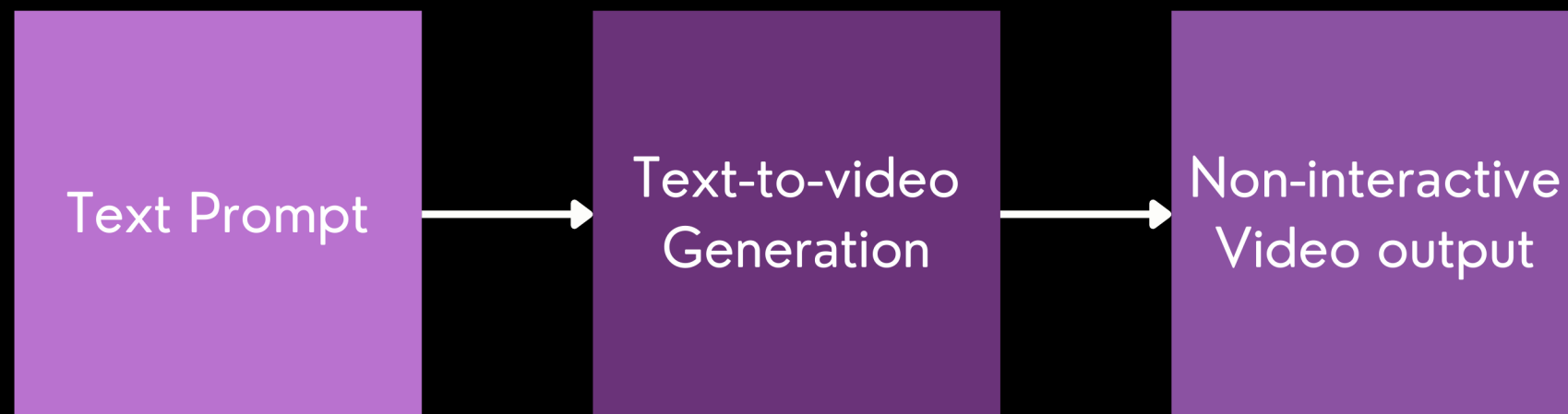
[5] Deepmotion - Video motion tracking



Issues: Inconsistent and jittering motion

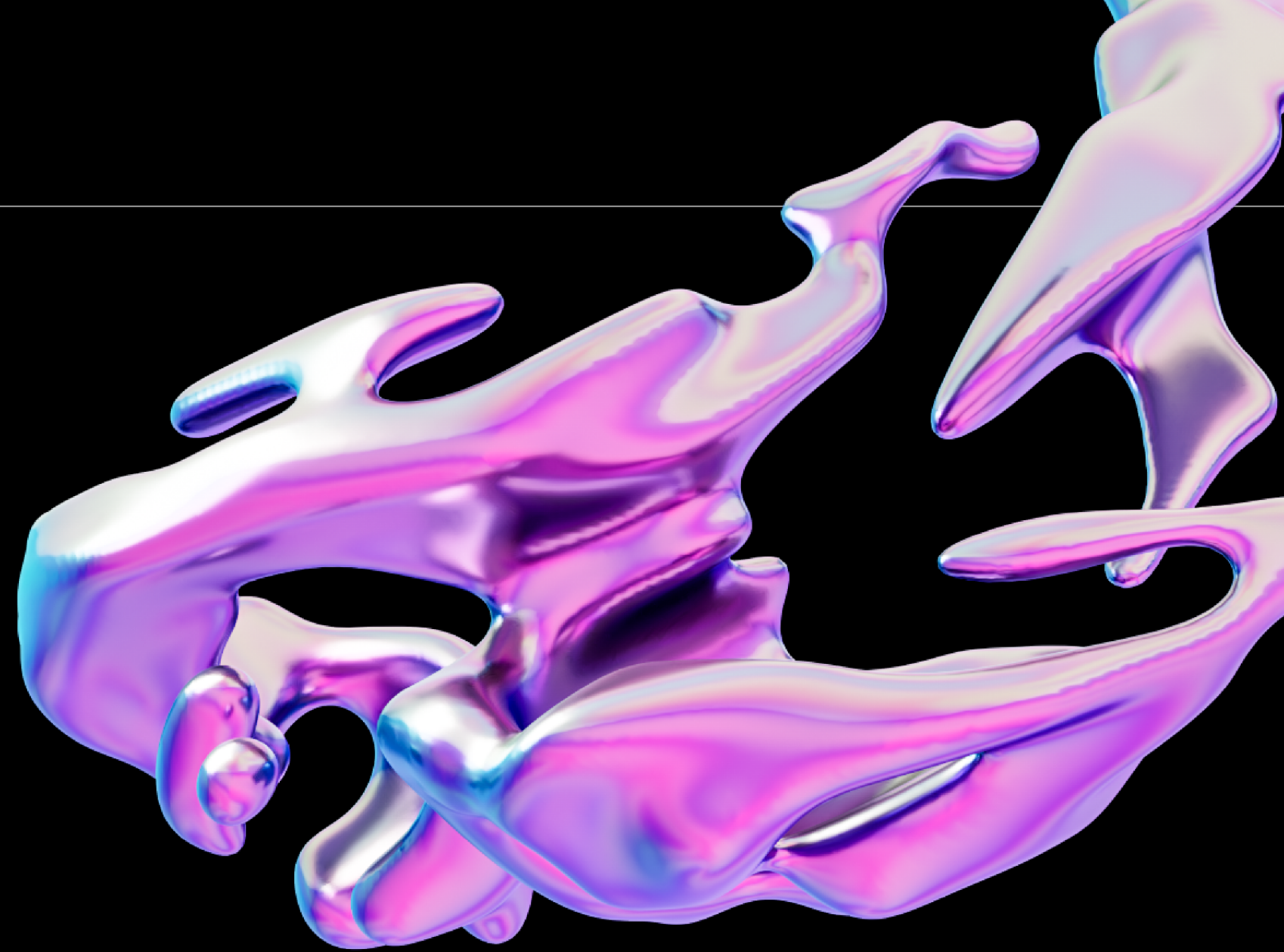
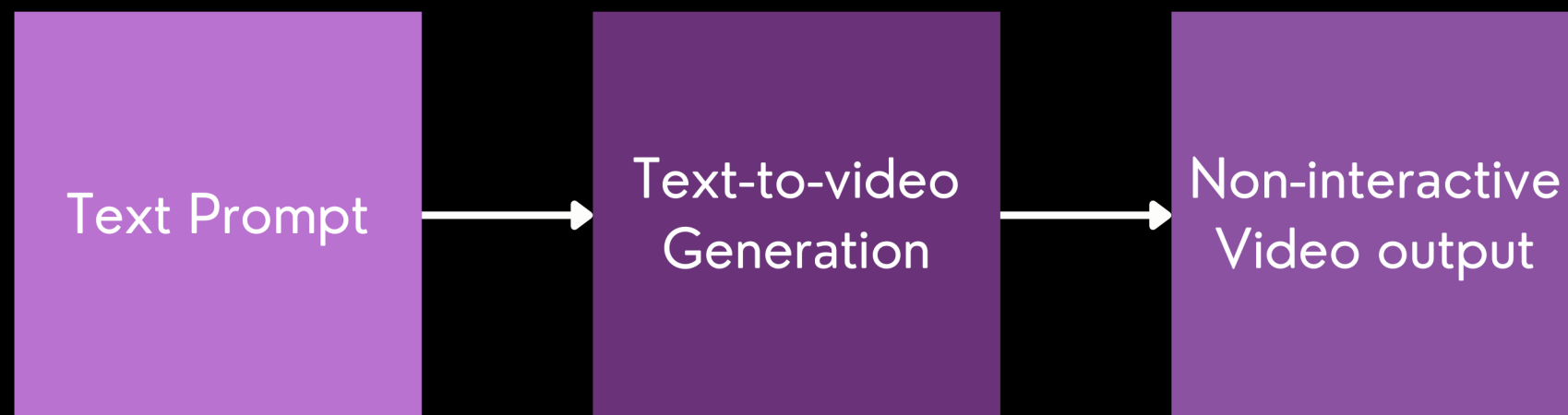
Motivation

Text-to-video generation Model



Motivation

Text-to-video generation Model

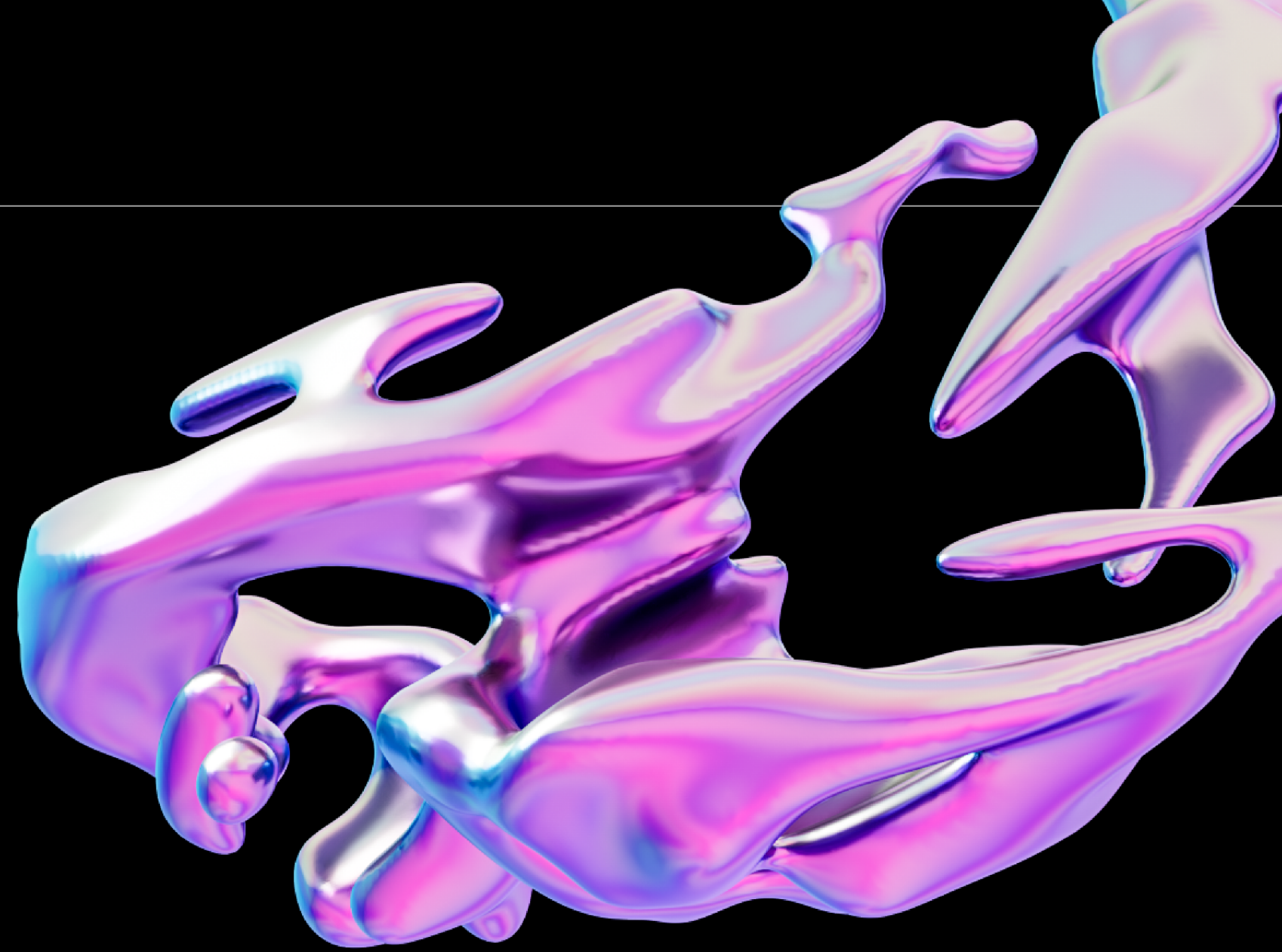


[6]



Motivation

Prompt: a man doing a backflip

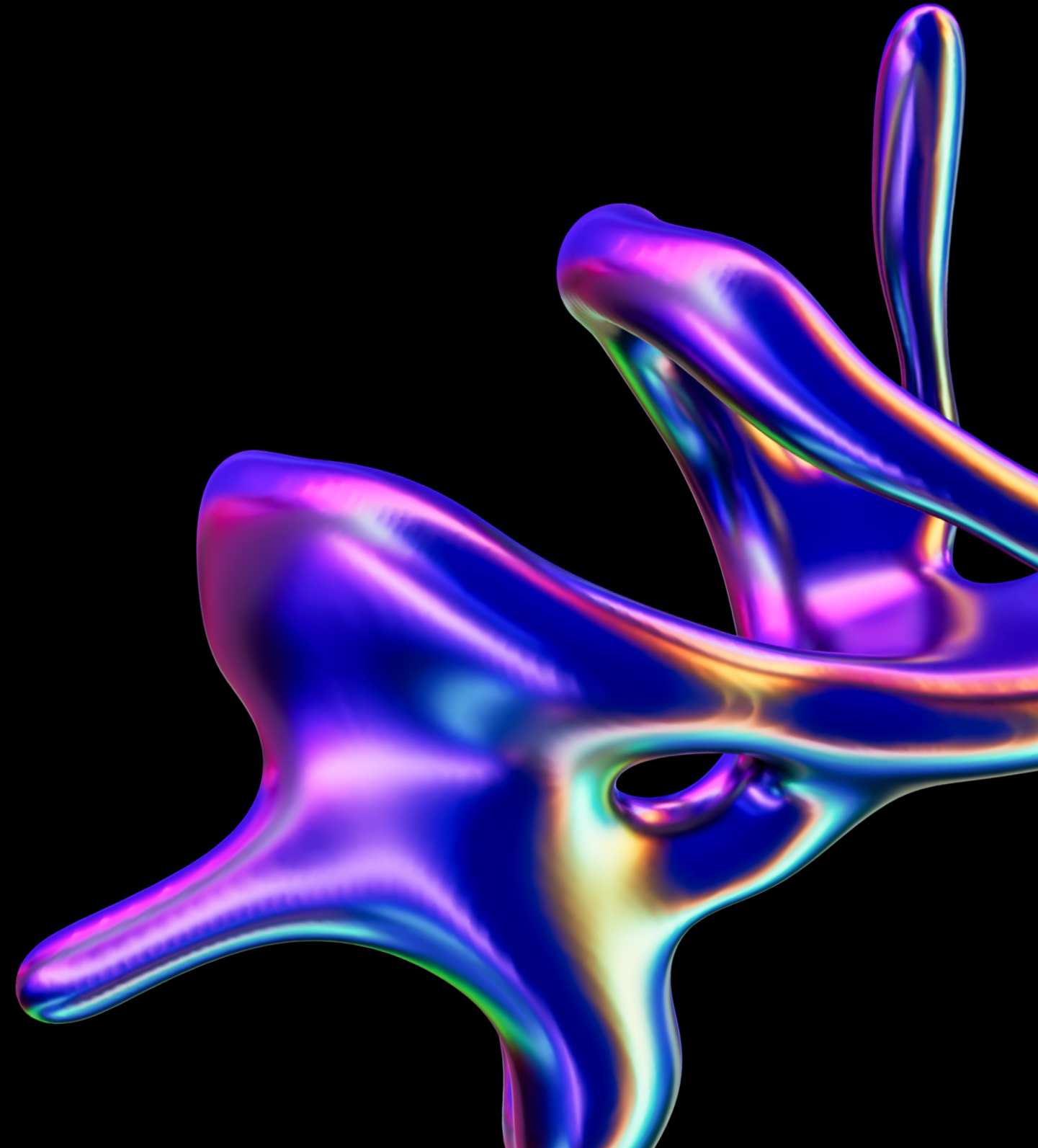


Motivation



For many text-to-video models:

- Format of the output is a non-interactive video
- Inconsistent frames and distorted figures
- Long generation time



Motivation



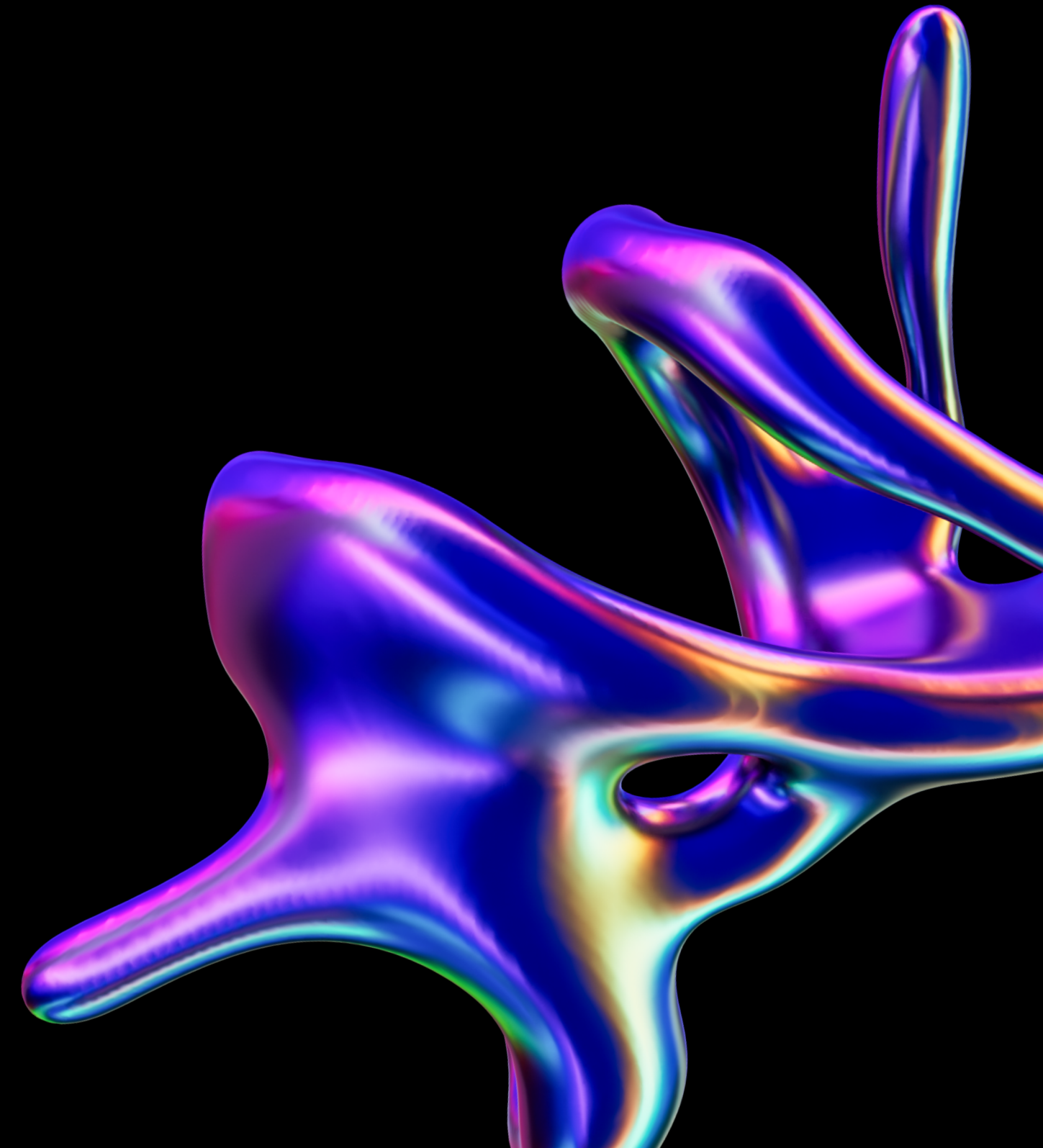
For many text-to-video models:

- Format of the output is a non-interactive video
- Inconsistent frames and distorted figures
- Long generation time

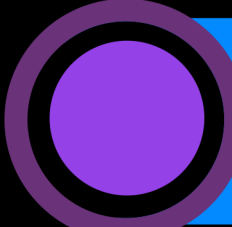
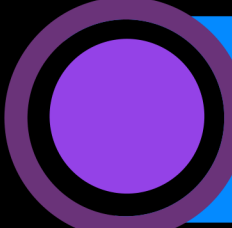
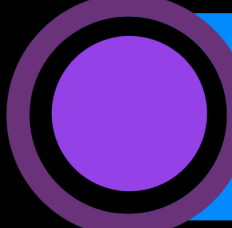
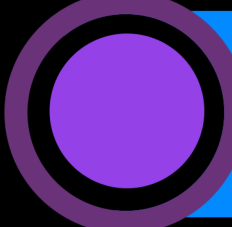


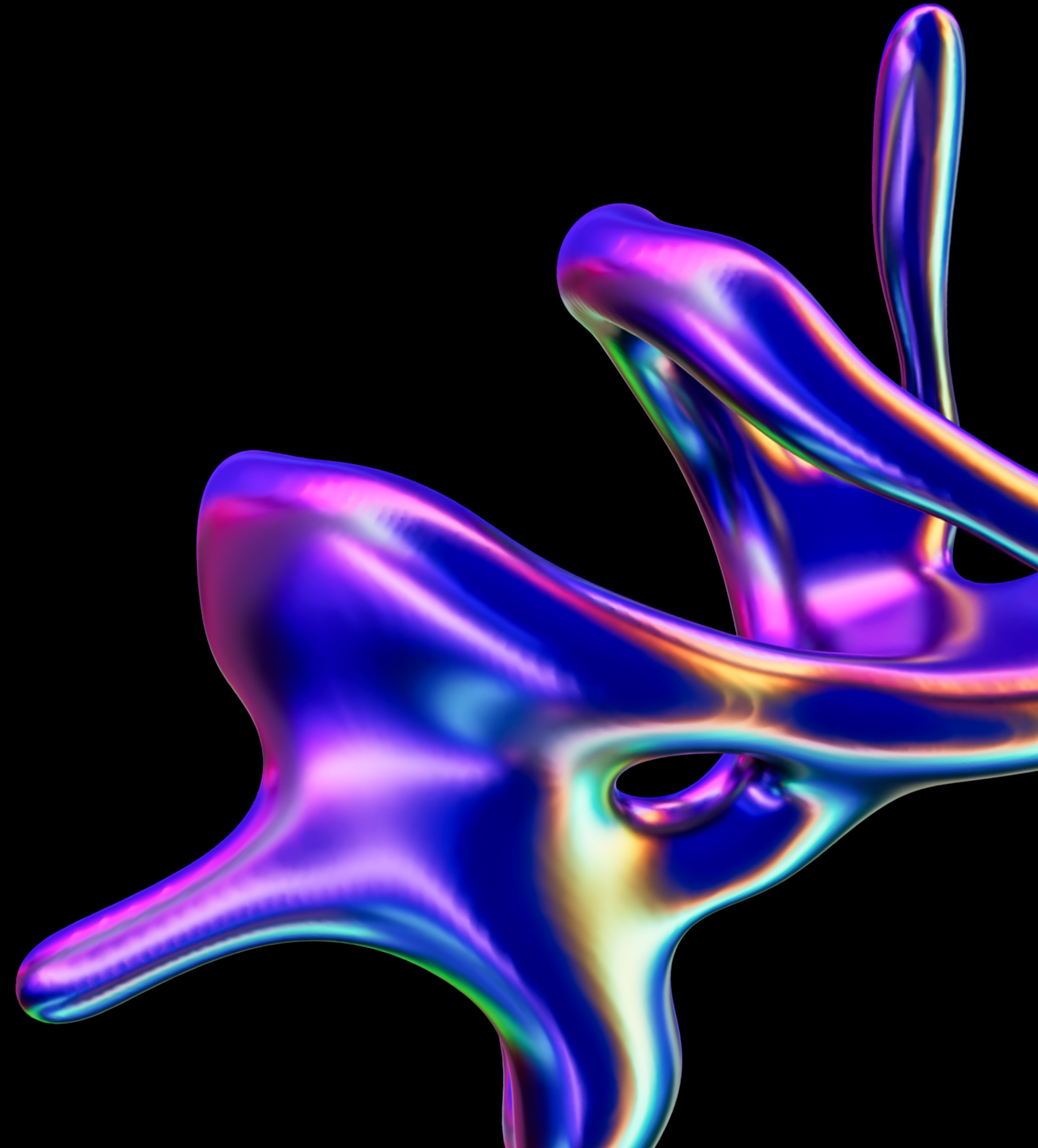
Changing up text-to-video models:

- 3D avatar movement in 3D scene
- Output will be in interactive format
- Does not require animation skills
- Consistent and production ready



Objectives

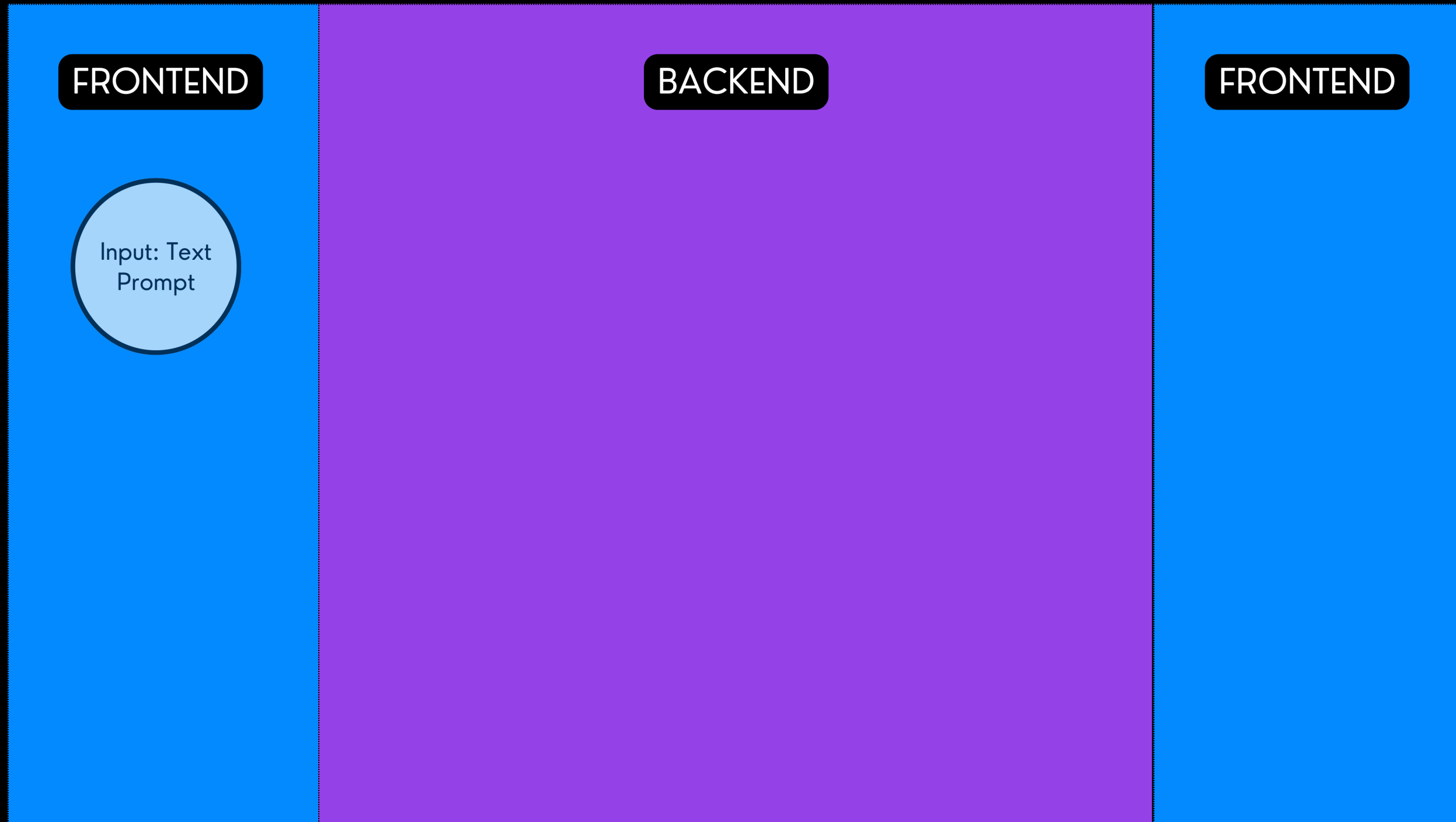
-  Used for Game Development and High Quality 3D Videos
-  Consistent output, no figure distortion, high quality
-  Output can be modified, editable, interactive
-  Easy to use web based interface



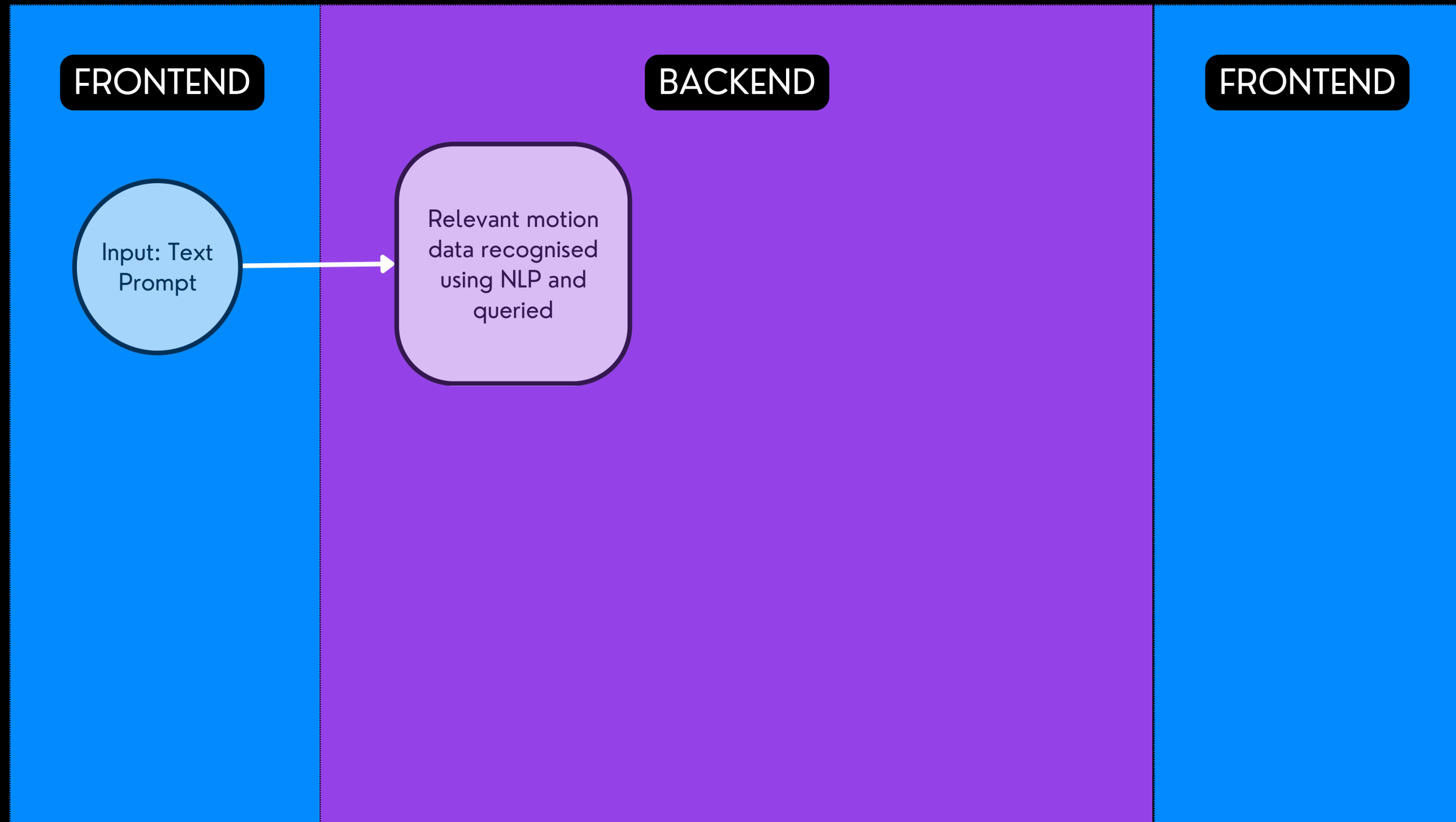
03. Methodology

- 03.1 Workflow
- 03.2 Tech stack
- 03.3 MVC Architecture

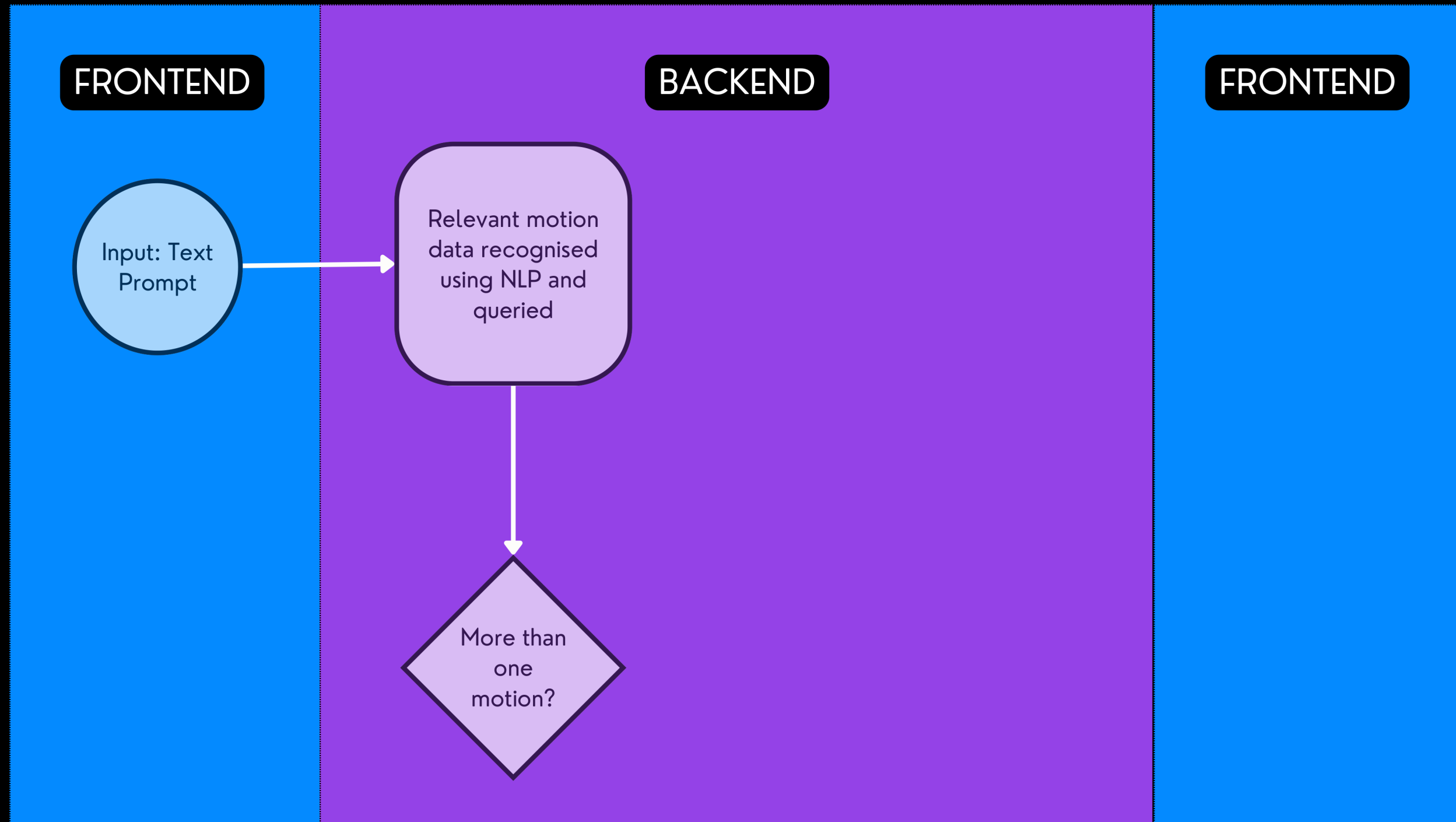




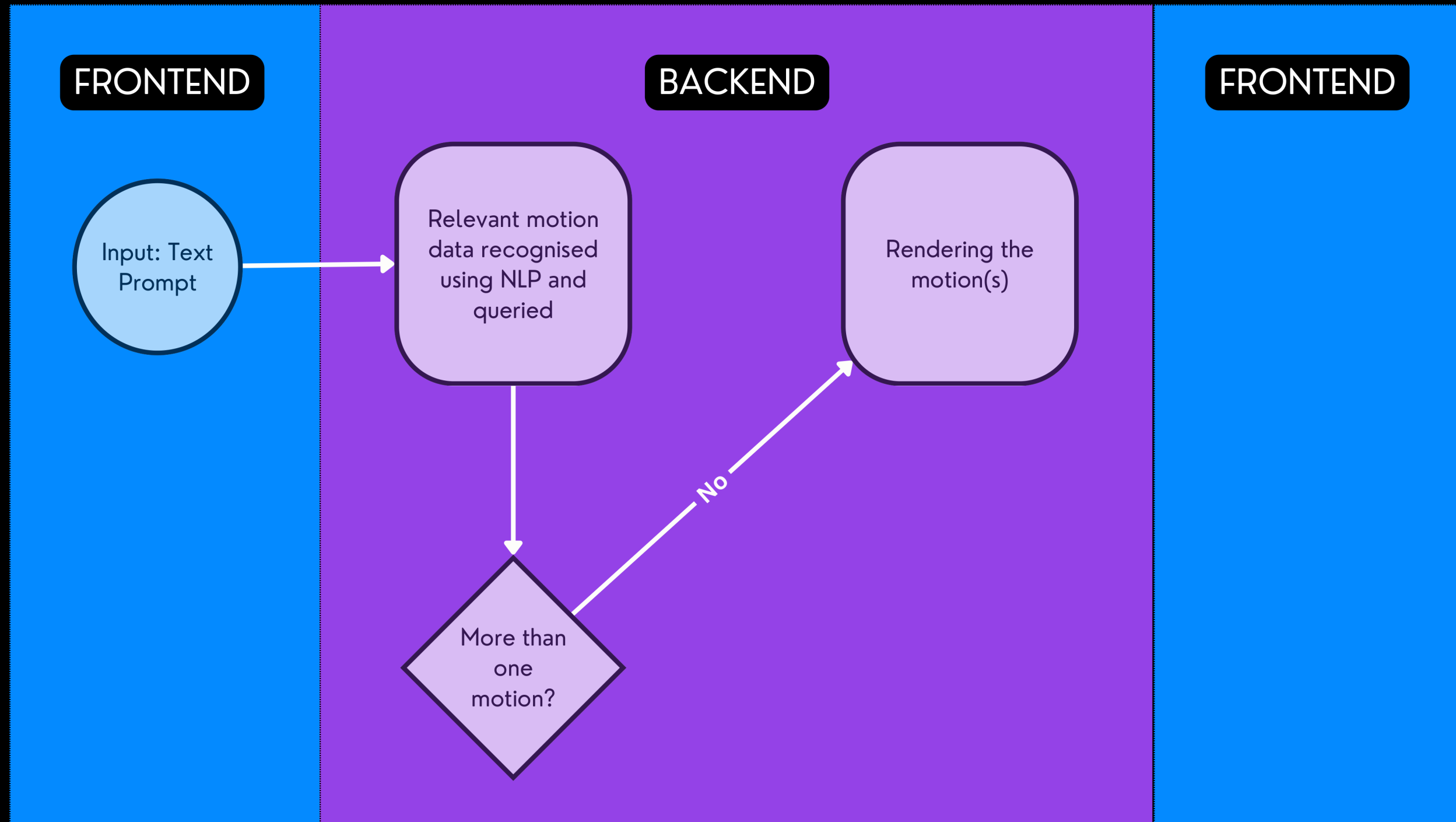
The Input-Process-Output workflow of AniGEN



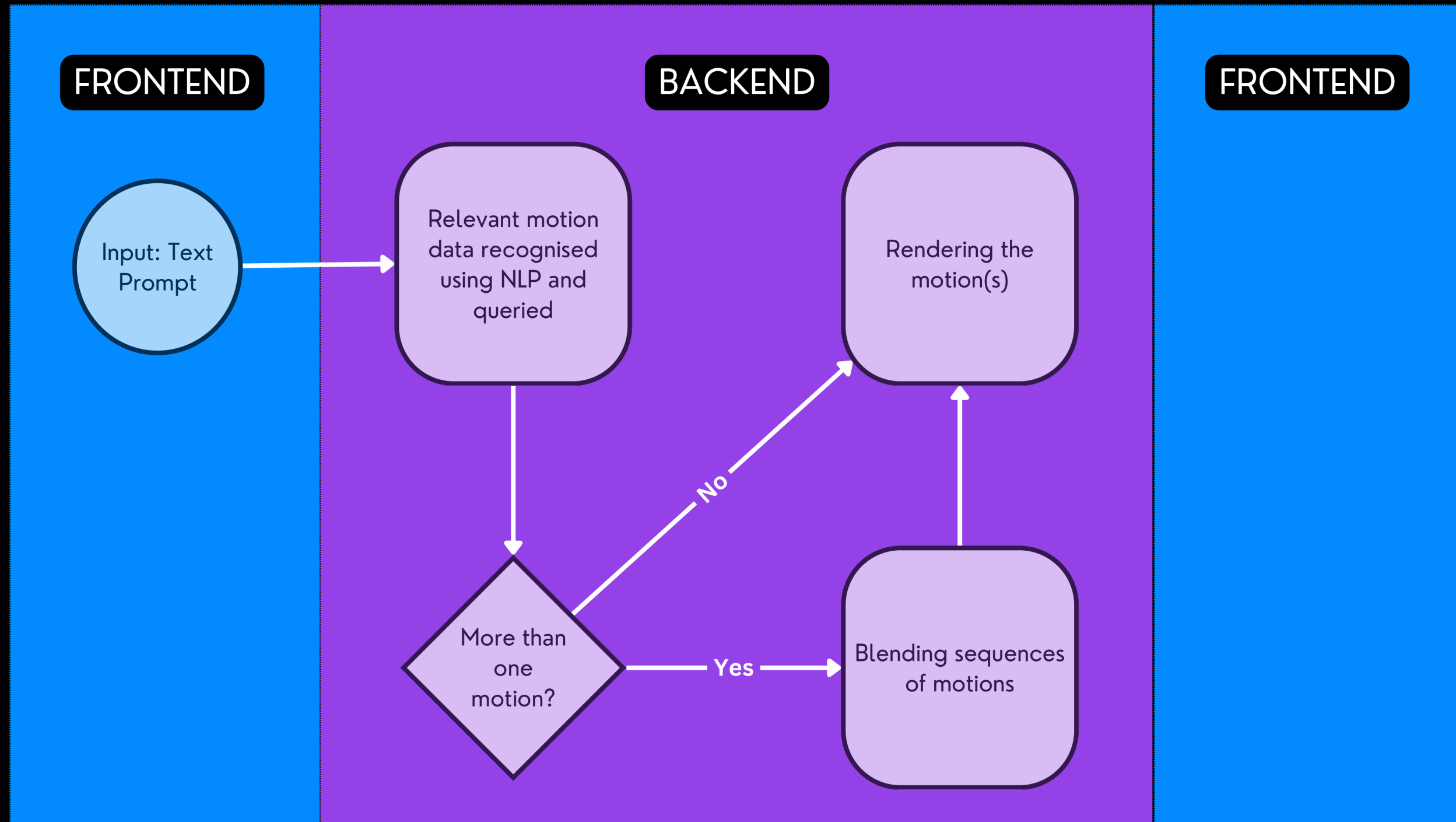
The Input-Process-Output workflow of AniGEN



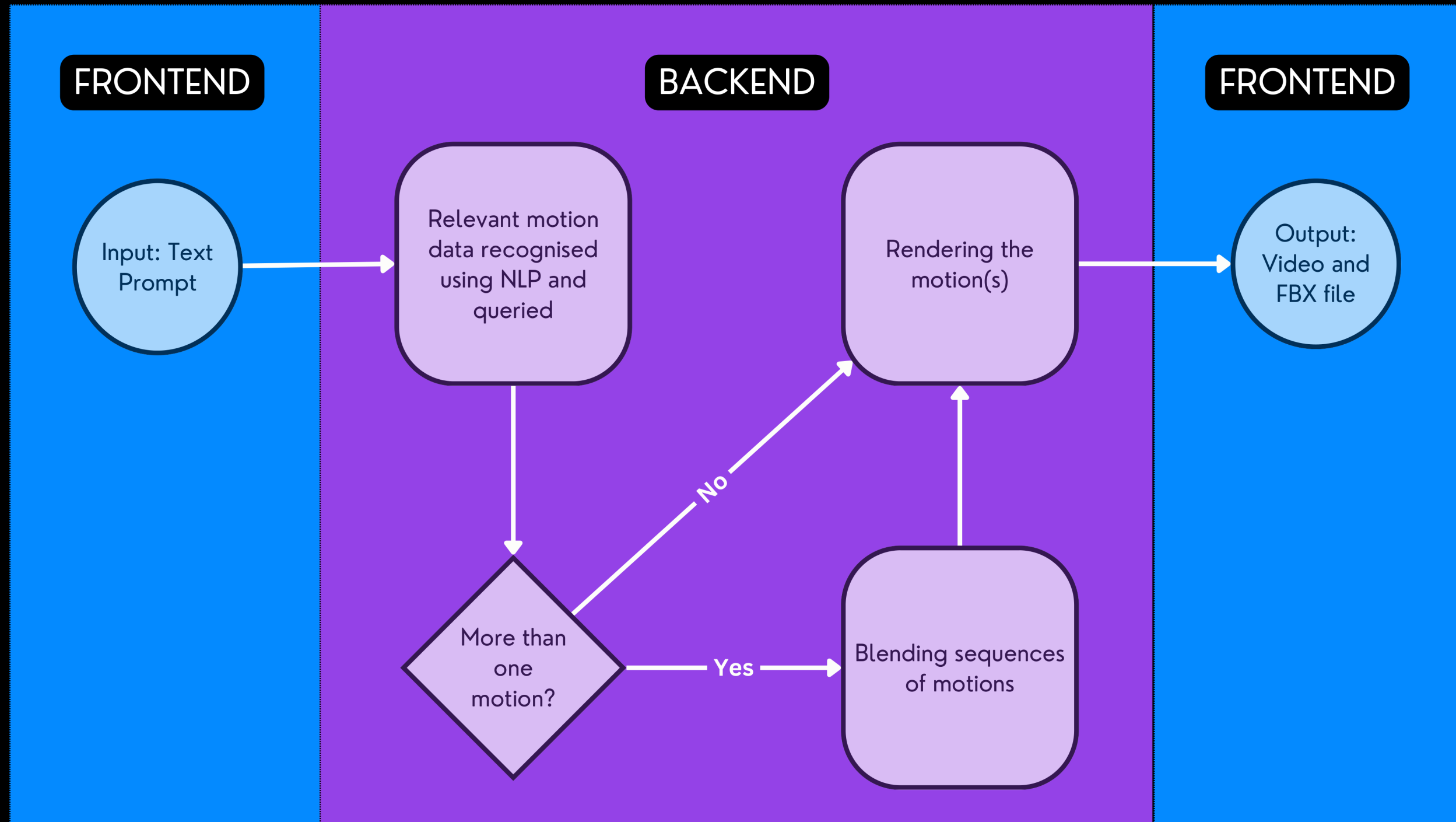
The Input-Process-Output workflow of AniGEN



The Input-Process-Output workflow of AniGEN

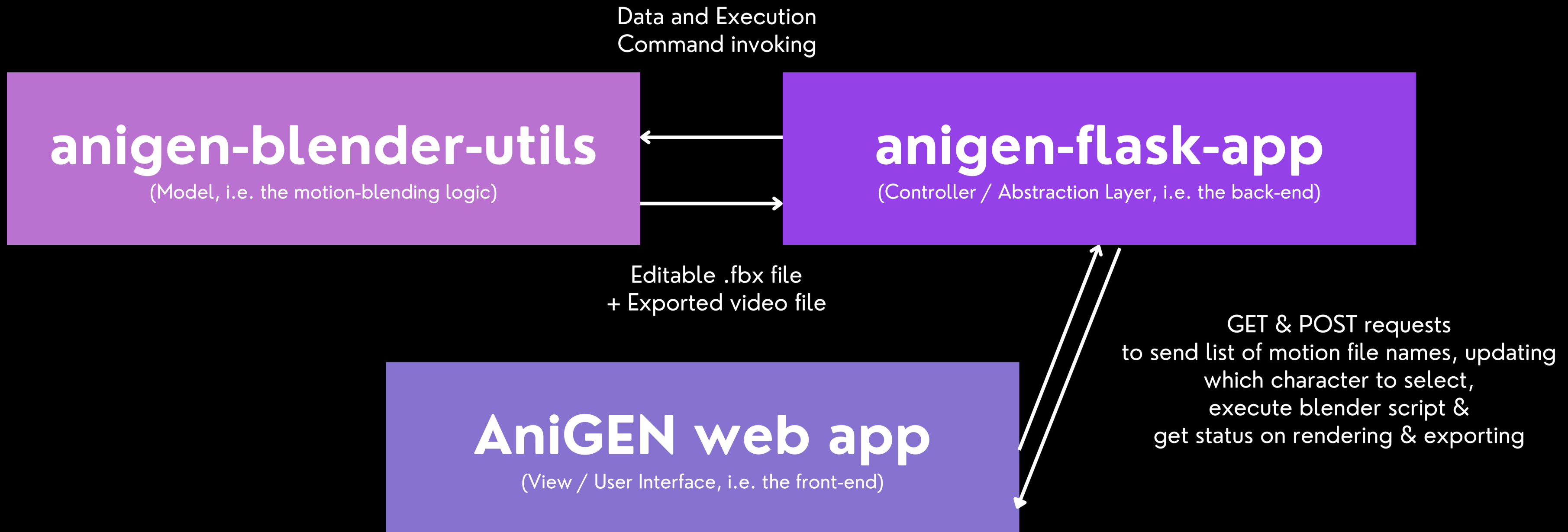


The Input-Process-Output workflow of AniGEN



The Input-Process-Output workflow of AniGEN

How AniGEN implements the Model-View-Controller architecture^[7]



Tech Stack

AniGEN webapp

React^[8] + **Vite**^[9]

(JavaScript Library & development server)

Bootstrap^[10]

(Front-end CSS Framework)

Tailwind CSS^[11]

(CSS Framework for simplifying styling)

Gemini^[12]

(Large Language Model API)

AniGEN-blender-utils

Blender^[13]

(Software for Motion and Rendering)

Mixamo^[14]

(Animation Data)

Python^[15]

(Scripting Language for Blender)

AniGEN-flask-app

Python

(Back-end code)

FireBase^[16]

(Database)

Flask^[17]

(Python Micro web framework)

Postman^[18]

(for testing API endpoints)



04. Final Results

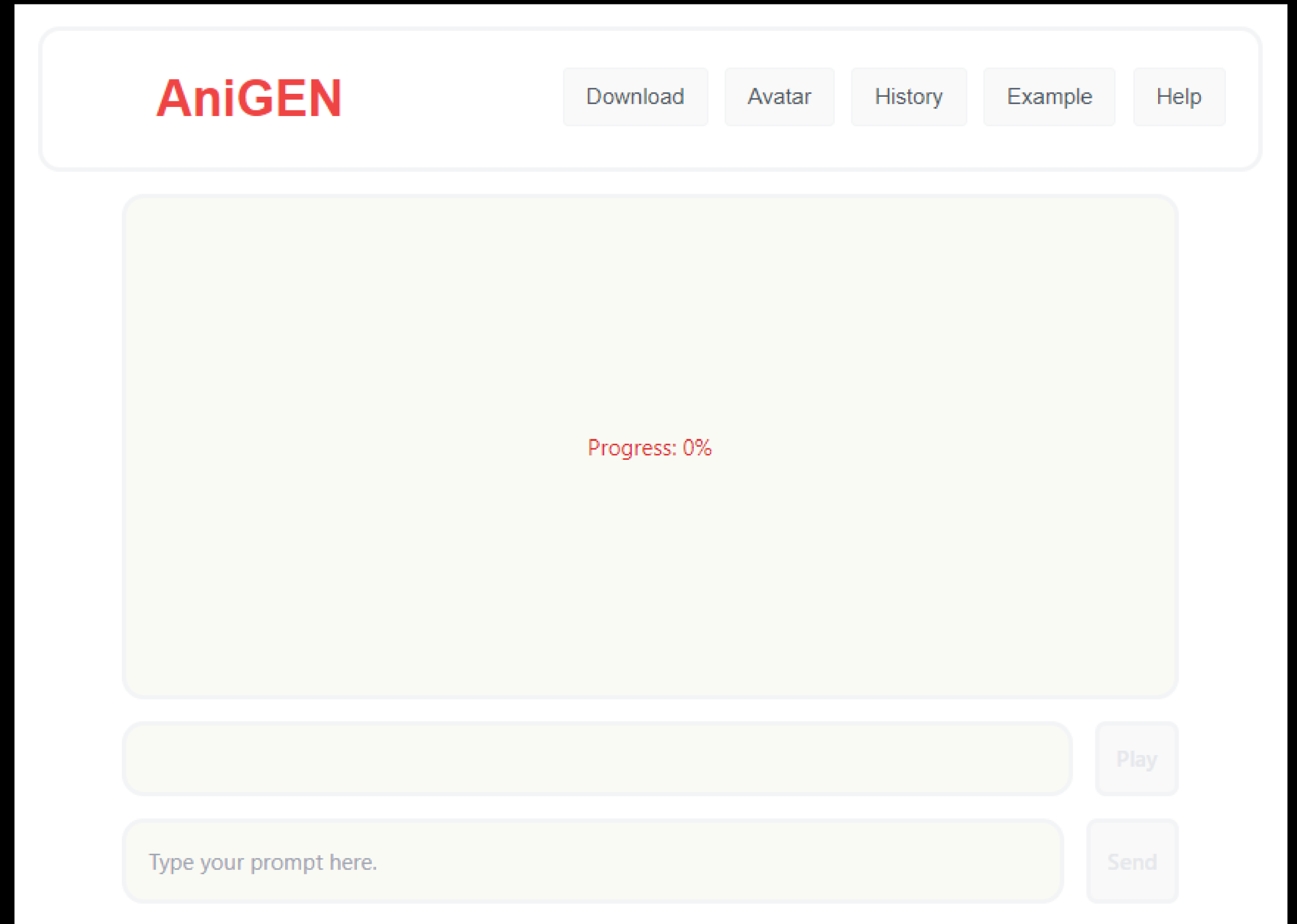
- 04.1 Web App
- 04.2 Blender Utils
- 04.3 Flask App
- 04.4 Difficulties
- 04.5 Technical Limitations

Web App Final Results

Final UI Design

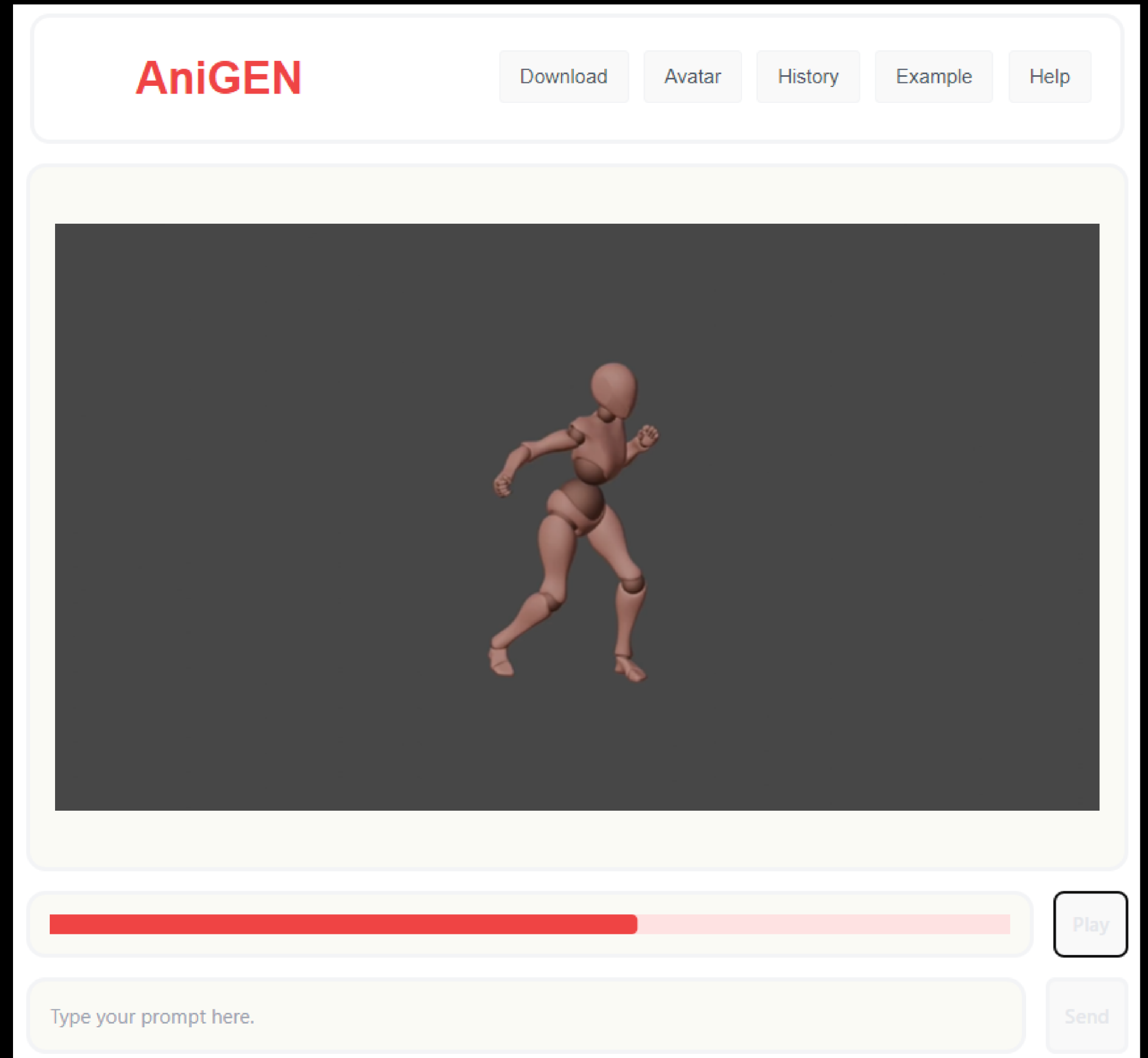
ChatGPT-like Interface

Simple and intuitive



Web App Final Results

What was changed?

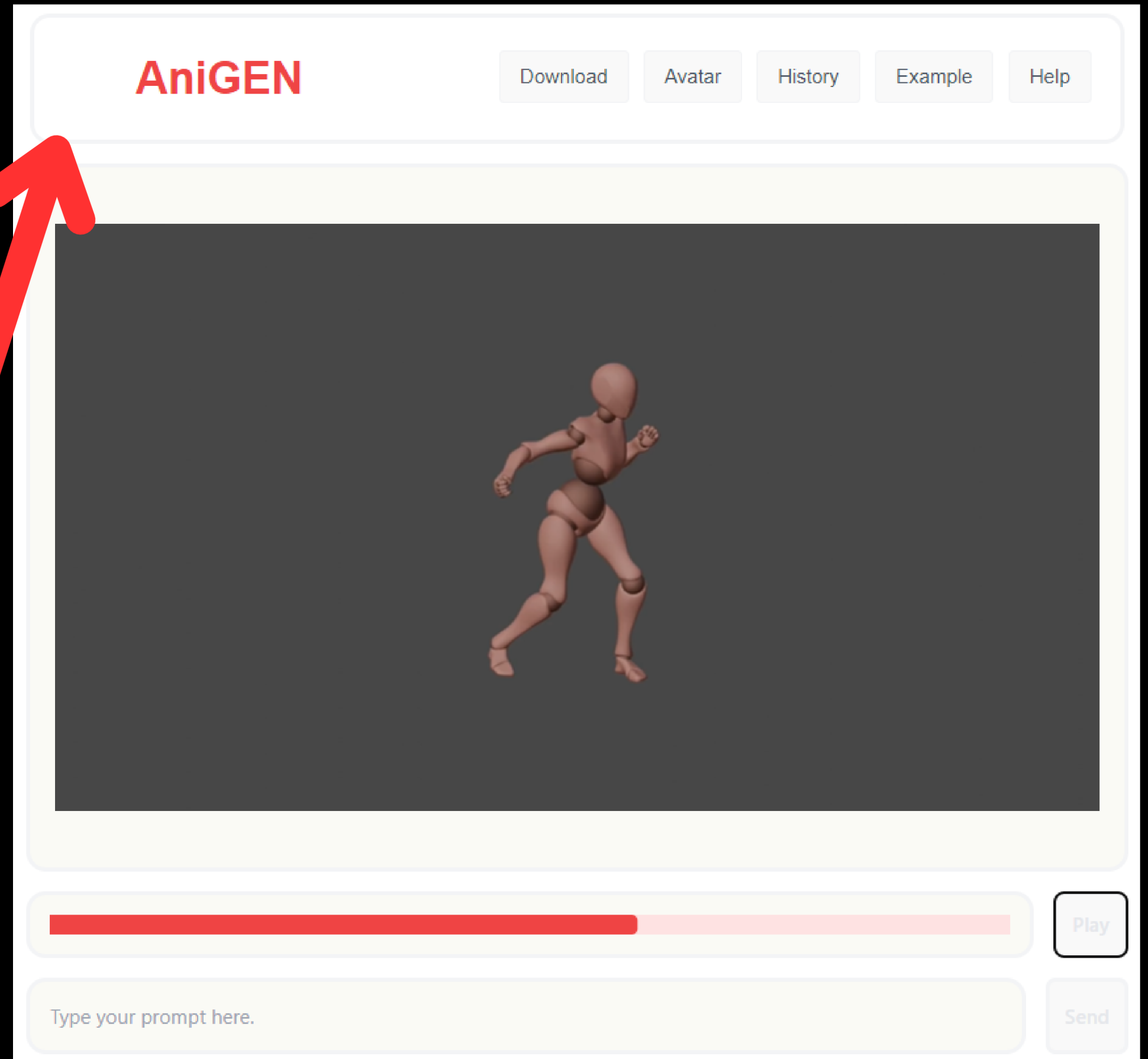


Web App Final Results

What was changed?

- Maximize space
- More compact

Navigation bar

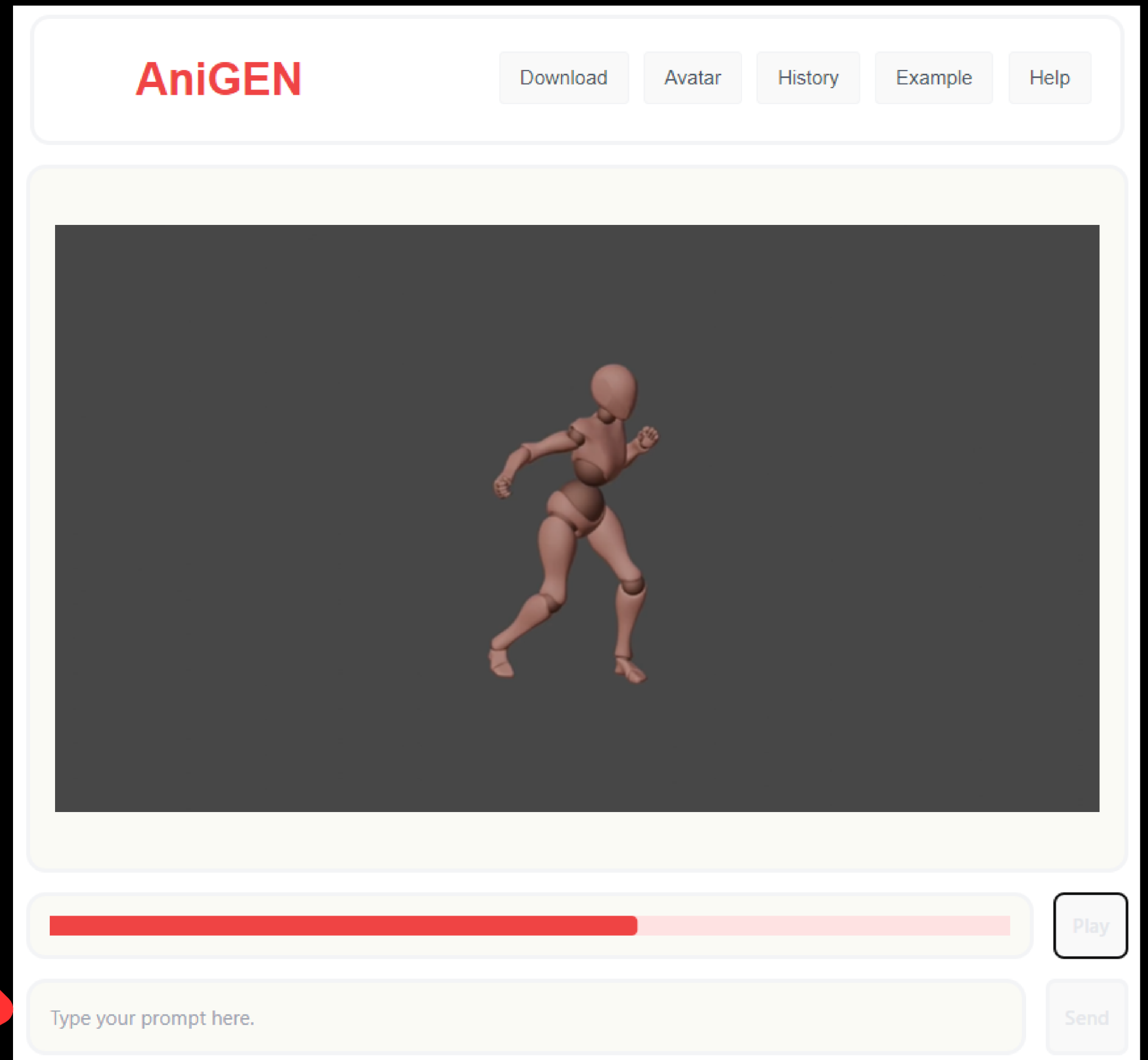


Web App Final Results

What was changed?

- Added a placeholder instruction
- Guide the users more clearly

Prompt box 

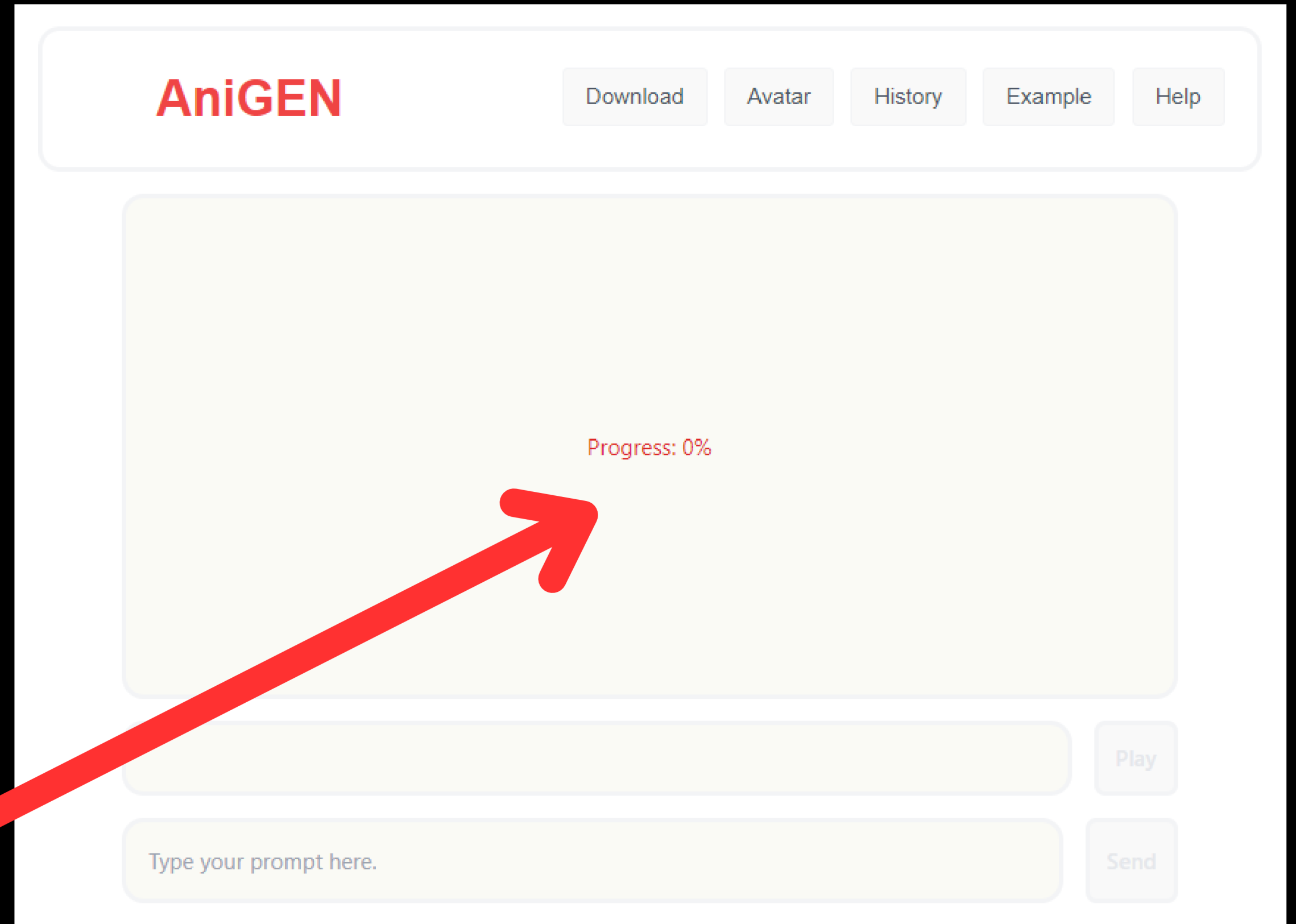


Web App Final Results

What was changed?

- Tells the user the progress
- Shows successful/failed rendering/exporting

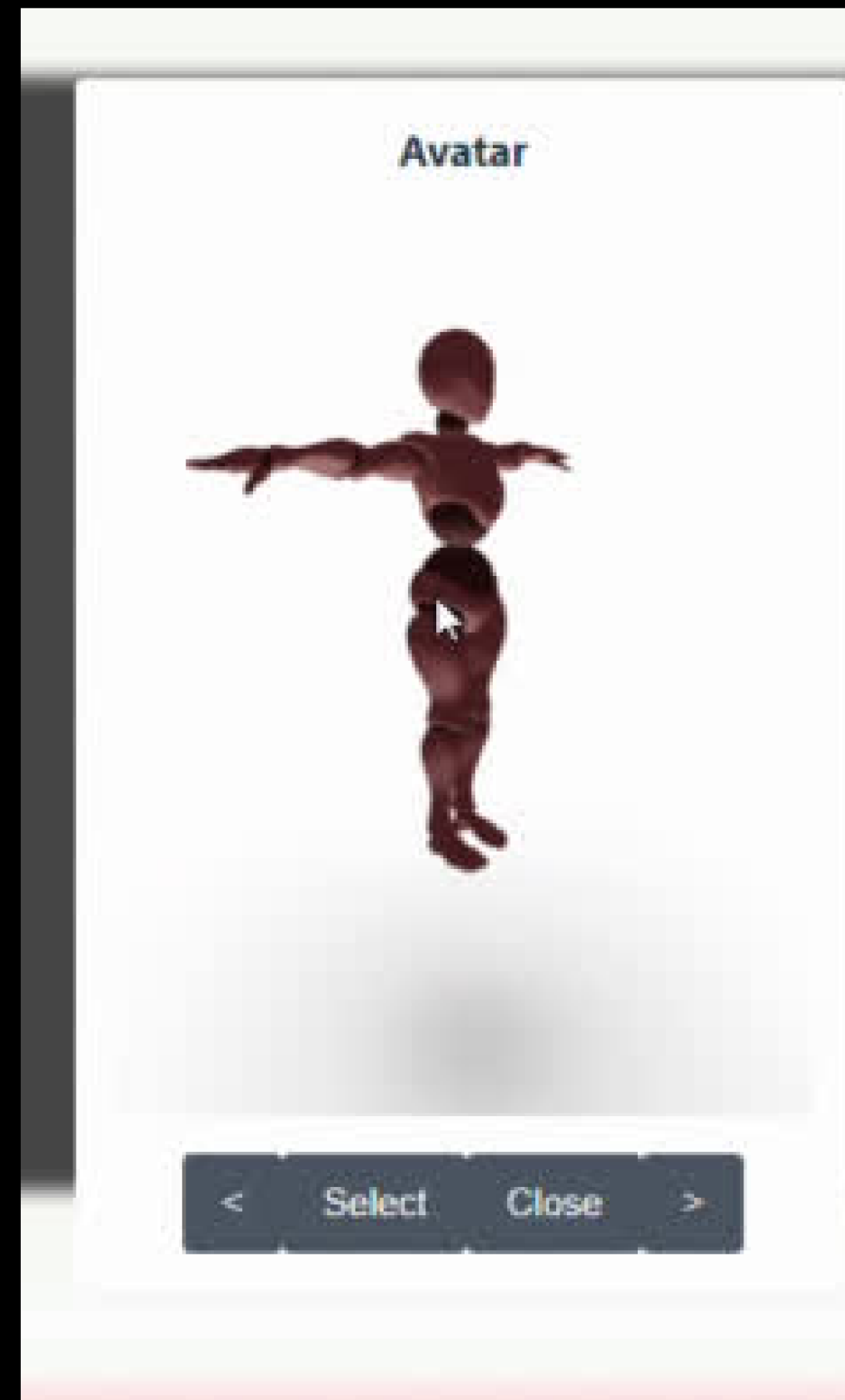
**Progress/Status
Indicator**



Web App Final Results

What was changed?

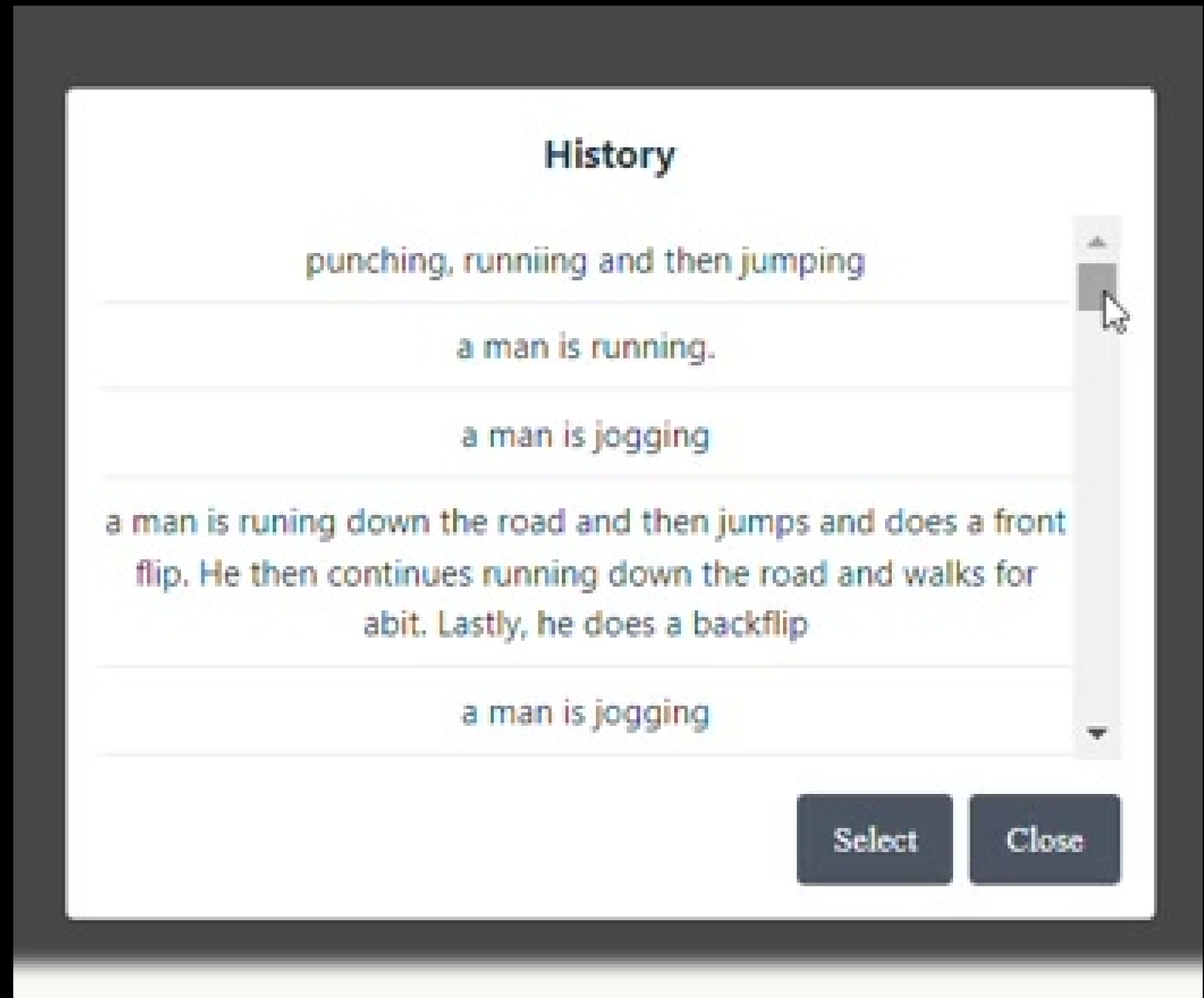
- Replaced with models used in the animation



Web App Final Results

What was changed?

- Prompt/History Persistence using Firebase



Web App Final Results

What was changed?

- Prompt Engineering through Gemini

```
const prompt =
```

```
`${prompt1}. Extract out they key motions in this sentence.
```

```
${filesString} This is the list of files.
```

```
Returns the only motion names that match the file names in the filesString
```

```
(One file for each motion if it is the same)
```

```
(If the sentence mentions a person's gender, try to find the file according to that gender).
```

```
Lastly, if there are repeated motions, make sure it is included again.
```

```
(Keep the order of the motions the same as the sentence.
```

```
Example: If a person is running and then jump and continues running,
```

```
the list of motions should be run, jump, run)
```

```
If there is none or one of the motions does not exist in the list of files, returns "null`;
```

Blender Utils Final Results

Implementing

- Motion Blending of In-Place Animation

For example,

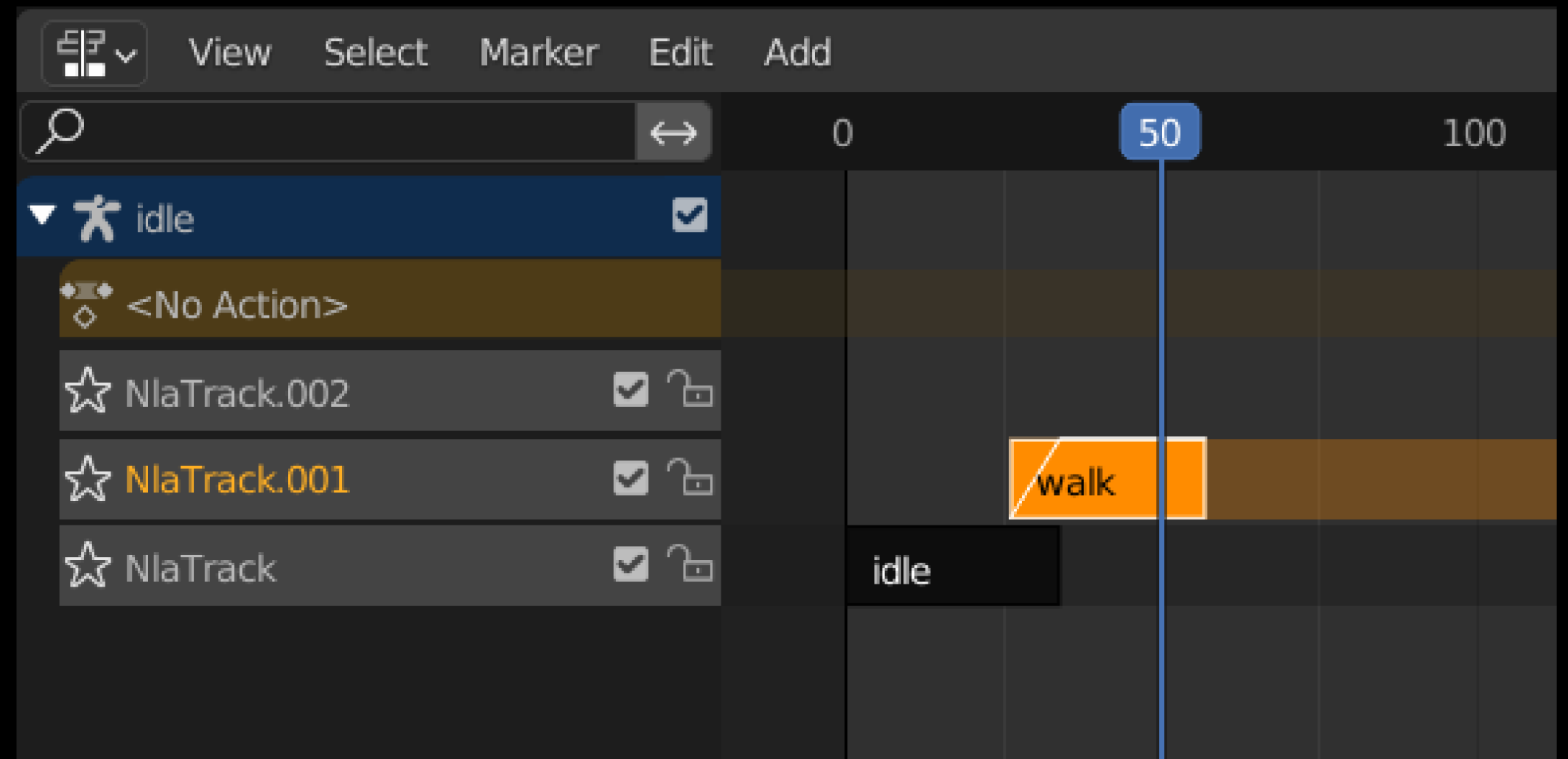
CREATING A TRANSITION OF
WALK TO RUN

Blender Utils Final Results

Implementing

- Motion Blending of In-Place Animation

Step 1: Put Motion 1 Clip in Blender

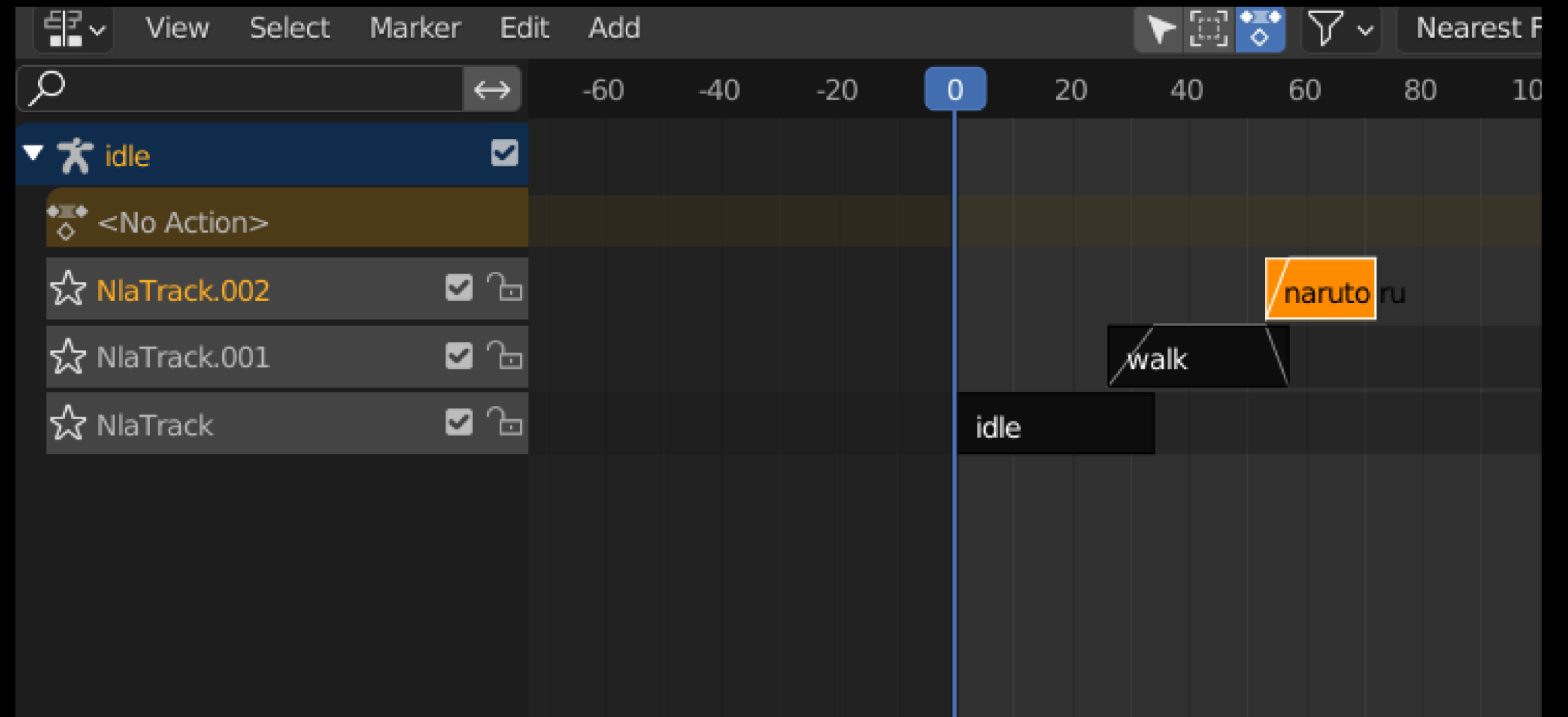


Blender Utils Final Results

Step 2: Put Motion 2 Clip in next track
above

Implementing

- Motion Blending of In-Place Animation

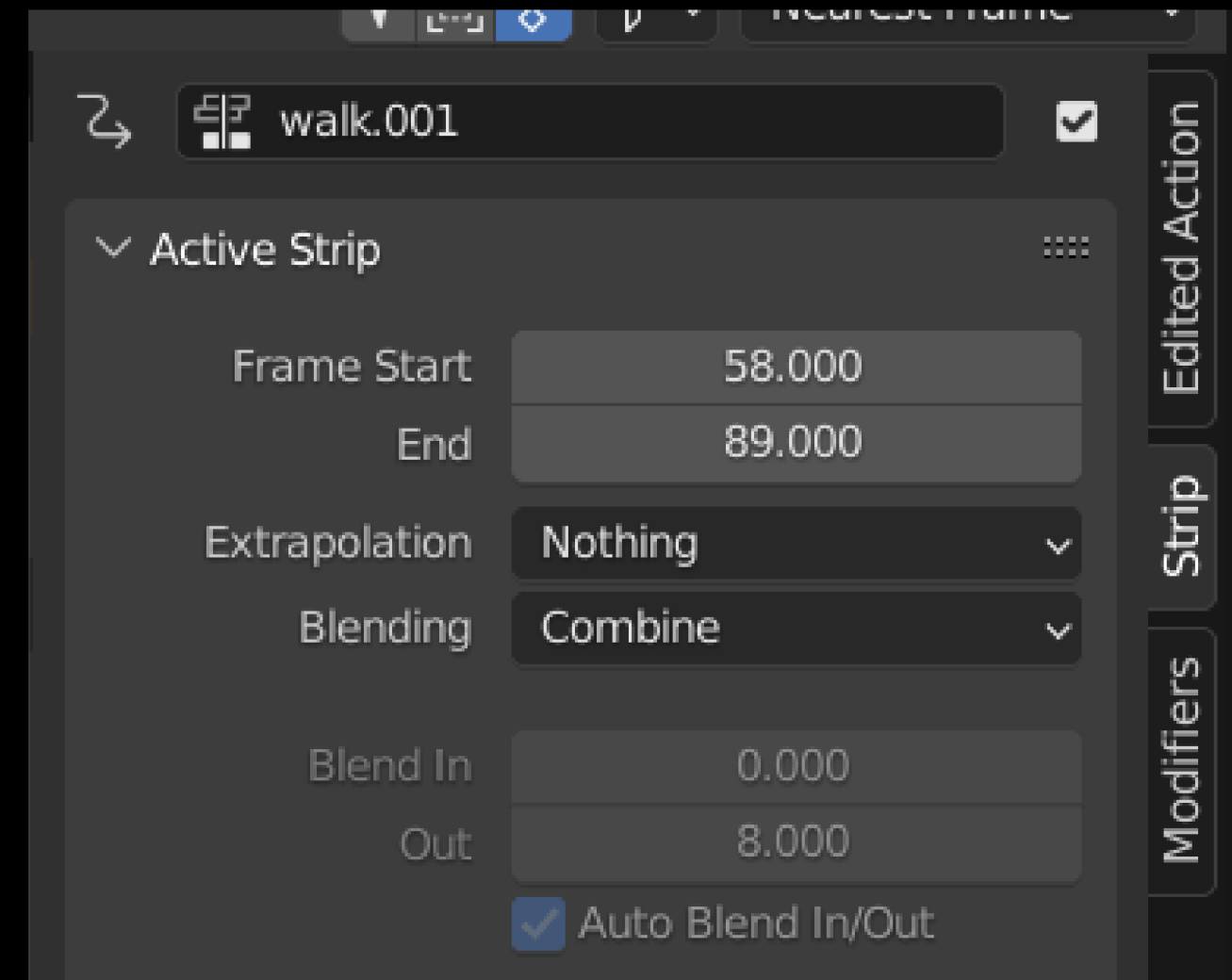


Blender Utils Final Results

Implementing

- Motion Blending of In-Place Animation

Step 3: Blend using Non Linear Animations^[18]



Blender Utils Final Results

Implementing

- Motion Blending of
In-Place Animation

Final Output:



Blender Utils Final Results

AniGEN-blender-utils

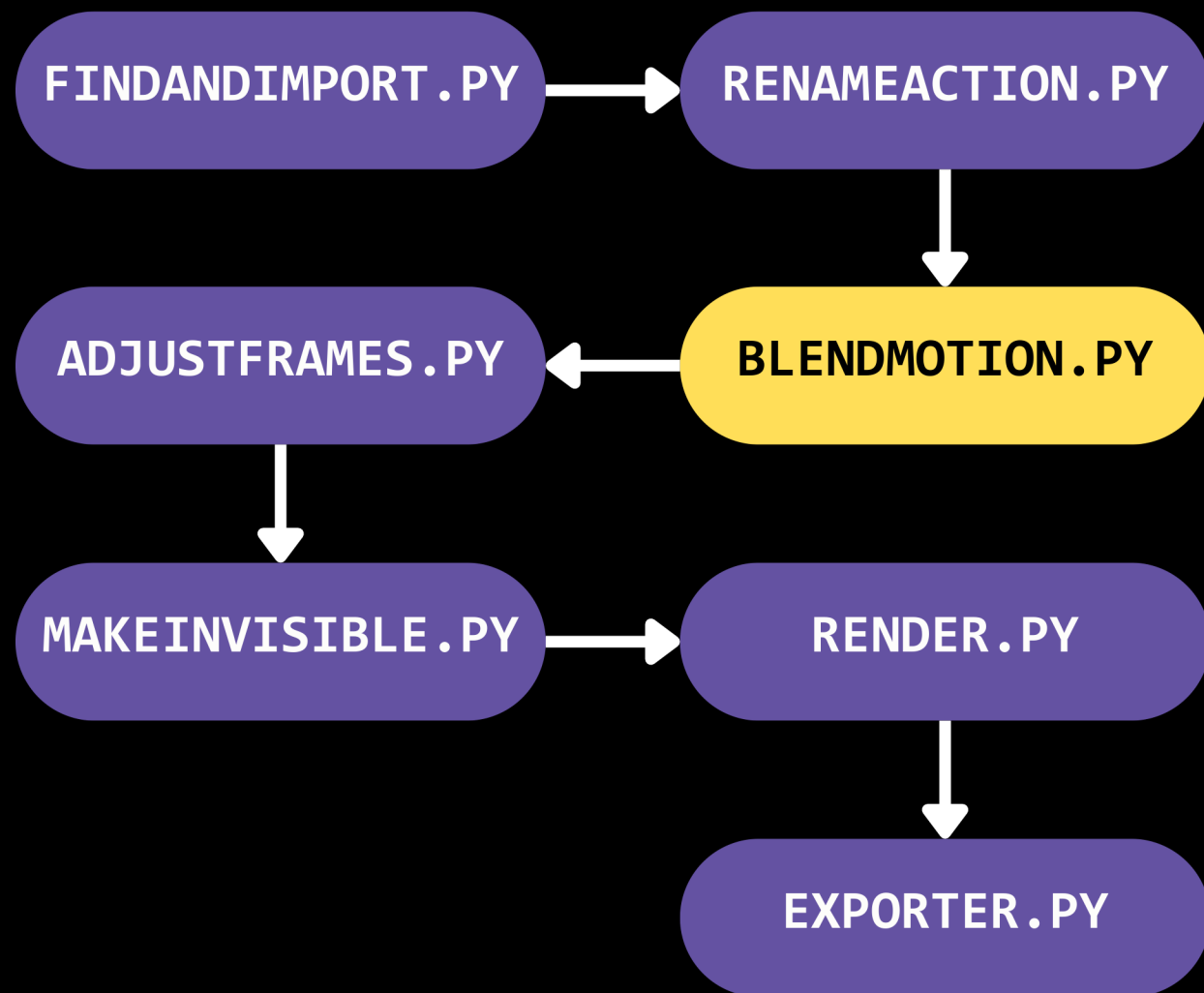
[BlendMotion.py]

Implementing

- Motion Blending of In-Place Animation

```
class BlendMotion:
    def run(self):
        """
        The main function that orchestrates the execution of the script
        Calls other functions to select the armature, retrieve the uppermost NLA track,
        get the last frame, switch to the NLA Editor, and push down the action to the NLA tracks
        """
        armature_obj = self.select_armature(self.armature_name)
        if armature_obj is not None:
            print(armature_obj.name)
            uppermost_track = self.get_uppermost_nla_track(armature_obj)
            self.switch_to_nla_editor()
            action = bpy.data.actions.get(self.action_name)
            if action is not None:
                armature_obj.animation_data.action = action
                self.push_down_action(armature_obj, uppermost_track, action)
            else:
                print("Action not found:", self.action_name)
```

Blender Utils Final Results



AniGEN-blender-utils

```

def main():
    # Import the blender file and the fbx files
    import_path = Config.IMPORT_PATH
    motions = Config.MOTIONS
    fbx_files = [f"{motion}.fbx" for motion in motions]
    find_and_import = FindAndImport(import_path, fbx_files)
    find_and_import.run()

    # Rename the actions and blend the motions for each of the motions
    for motion in motions:
        rename_action = RenameAction(motion)
        rename_action.run()
    for motion in motions:
        blend_motion = BlendMotion("idle", motion)
        blend_motion.run()

    # Fix the strips timings for all the NLA tracks
    adjust_frames = AdjustFrames("idle", offset=10)
    adjust_frames.fix_strips_timings()

    # Make the armature objects invisible in the outliner
    make_invisible = MakeInvisible()
    make_invisible.run()
    # Render the animation
    render_path = Config.RENDER_PATH
    renderer = Render(...)
    renderer.run()
    # Export the armature as FBX
    export_fbx = Exporter(...)
    export_fbx.run()
  
```

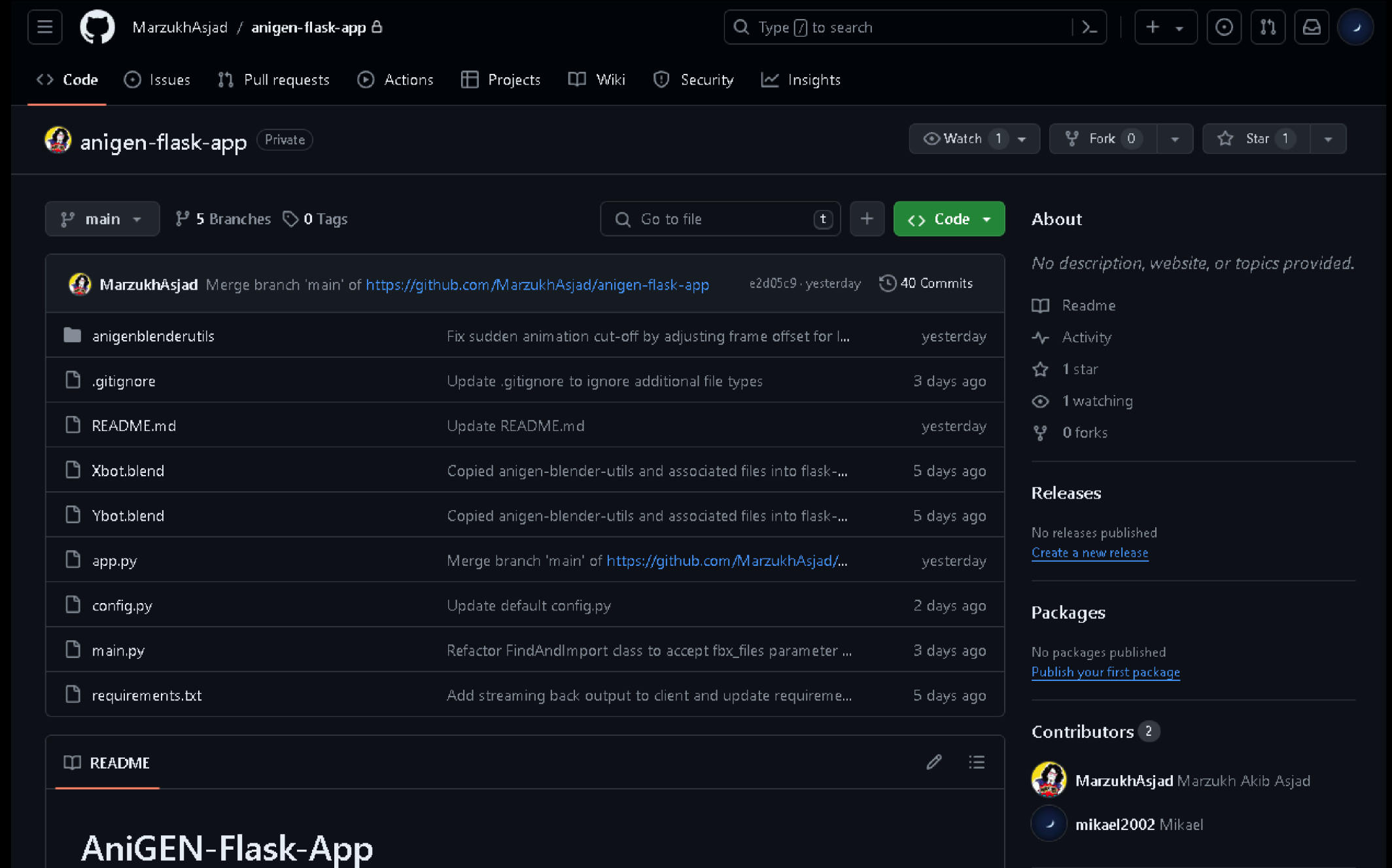

Flask App Final Results

Endpoint at:



Implemented

- App written with the Flask framework to facilitate communication between web app and blender-utils



The screenshot shows a GitHub repository page for 'anigen-flask-app' by MarzukhAsjad. The repository is private and has 1 star, 0 forks, and 1 watch. The main branch is selected, showing 5 branches and 0 tags. The repository contains 40 commits. The commit history is as follows:

Commit	Message	Time
MarzukhAsjad	Merge branch 'main' of https://github.com/MarzukhAsjad/anigen-flask-app	e2d05c9 - yesterday
	anigenblenderutils	Fix sudden animation cut-off by adjusting frame offset for l... yesterday
	.gitignore	Update .gitignore to ignore additional file types 3 days ago
	README.md	Update README.md yesterday
	Xbot.blend	Copied anigen-blender-utils and associated files into flask-... 5 days ago
	Ybot.blend	Copied anigen-blender-utils and associated files into flask-... 5 days ago
	app.py	Merge branch 'main' of https://github.com/MarzukhAsjad/... yesterday
	config.py	Update default config.py 2 days ago
	main.py	Refactor FindAndImport class to accept fbx_files parameter ... 3 days ago
	requirements.txt	Add streaming back output to client and update requireme... 5 days ago

The README file is titled 'AniGEN-Flask-App'. The repository also includes sections for About, Releases, Packages, and Contributors. The About section states 'No description, website, or topics provided.' The Releases section indicates 'No releases published' and provides a link to 'Create a new release'. The Packages section indicates 'No packages published' and provides a link to 'Publish your first package'. The Contributors section lists two contributors: MarzukhAsjad (Marzukh Akib Asjad) and mikael2002 (Mikael).

Flask App Final Results

Endpoint at:

/motions

Implemented

- Method to receive "POST" from web app with json containing names of motions and to update the config file with the received data

```
17 # This method will receive a json which will contain names of motions
18 @app.route('/motions', methods=['POST'])
19 def motions_receive():
20     data = request.json
21     # Store motions in the config file
22     Config.MOTIONS = data['motions']
23     # Append the last motion to the Config.MOTIONS 4 times and smooth it to 'idle'
24     Config.MOTIONS = Config.MOTIONS + [Config.MOTIONS[-1]] * 4 + ['idle']
25     write_config_file()
26     return '', 200
```

Flask App Final Results

Endpoint at:

/character

Implemented

- Method to receive "POST" from web app with json containing info about blender character and to update the config file with the received data

```
28 # This method will receive a json which will contain the information about the blender character
29 @app.route('/character', methods=['POST'])
30 def character_receive():
31     # Extract the character name from the json
32     data = request.json
33     character = data['character']
34     # Store character information in the config file's BLEND_PATH
35     blend_path = r'C:\Users\User\Desktop\FYP\flask-app\{}.blend'.format(character)
36     Config.BLEND_PATH = blend_path
37     write_config_file()
38     return '', 200
39
```


Flask App Final Results

Endpoint at:

/notification

Implemented

- Method to receive "GET" from web app, responding the web app with code and status as a json response.

```
40 # The notification receiver
41 @app.route('/notification', methods=['GET'])
42 def notification():
43     # Extract the code and status from the config file
44     # Return the code and status as a json response
45     payload_cs = jsonify({'code': Config.CODE, 'status': Config.STATUS})
46     payload_cs.headers['Access-Control-Allow-Origin'] = '*'
47     return payload_cs, 200
```

'Access-Control-Allow-Origin' header of payload set to '*'
to allow web app to display code and status in frame



Flask App Final Results

Endpoint at:

/exec

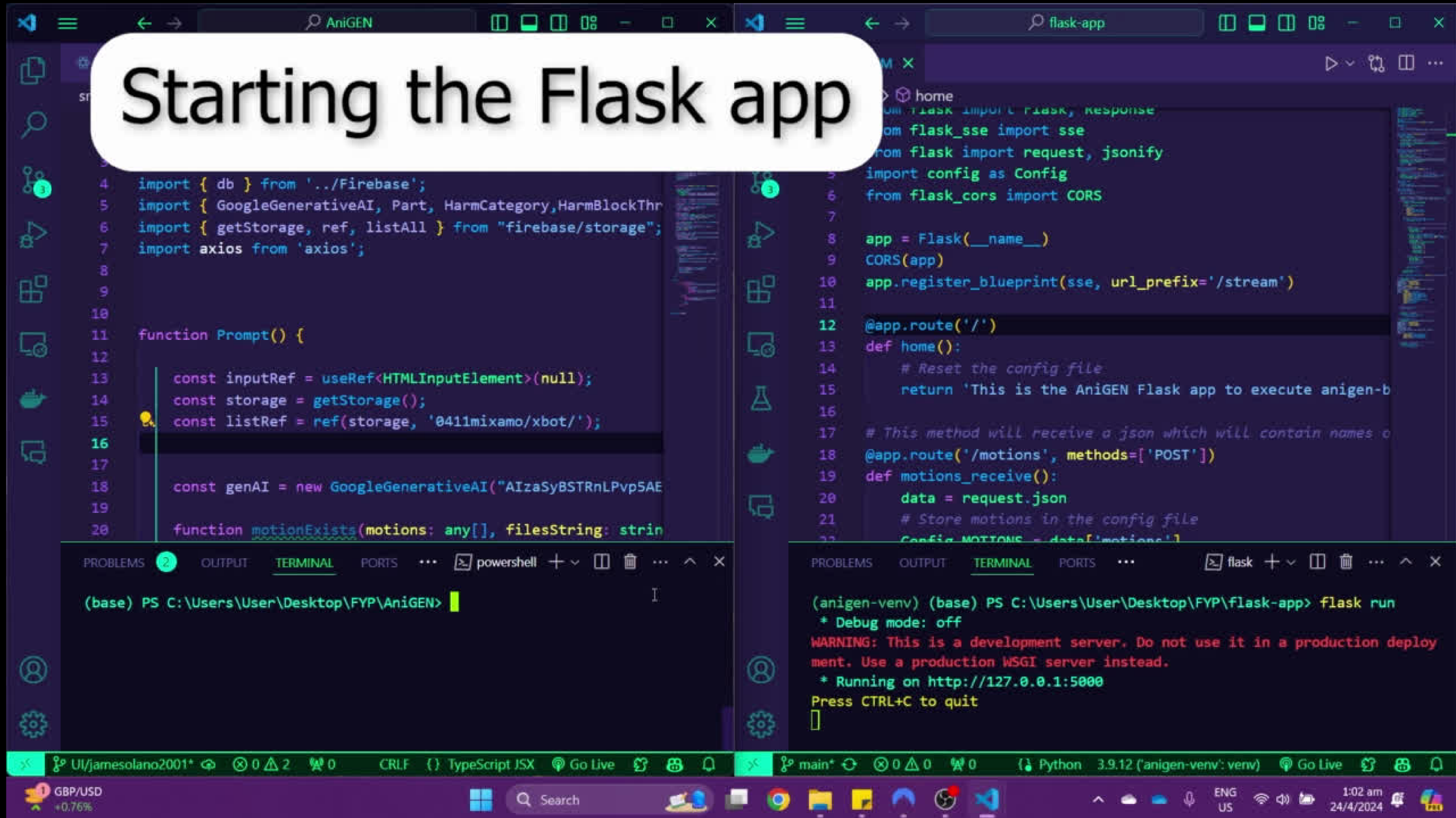
Implemented

- Method to receive "GET" from web app to execute the blending and rendering script which also streams the output of the command and processes it

```
54 @app.route('/exec')
55 def execute_command():
56     # Create a file to store the log
57     log = open('log.txt', 'w')
58     # The command to be executed
59     command = r'blender {} --background --python main.py'.format(Config.BLEND_PATH)
60     # Function to stream the output of the command back to the client
61     def stream_output():
62         nFrames = Config.TOTAL_FRAMES
63         # Execute the command by creating a subprocess and reading the output
64         process = subprocess.Popen(...
72     # Read the output line by line
73     for line in iter(process.stdout.readline, ''):
74         # Process the output to get the necessary information
75         yield line.rstrip() + '\n'
76         # Render start
77         if line.startswith('Blender ') and 'quit' not in line: ...
80         # Frame progress
81         elif line.startswith('Append frame '):
82             iFrame = int(line.removeprefix('Append frame '))
83             percent_progress = int((iFrame * 100) / nFrames)
84             # Render progress
85             Config.CODE = 'P'
86             Config.STATUS = percent_progress
87             # Render complete
88             if iFrame == nFrames: ...
95         # Export complete
96         elif line.startswith('Export complete!'): ...
101         # Write the output to the log file
102         log.write(line)
103         # Close the log file and the process
104         process.stdout.close()
105         process.wait()
106
107     return Response(stream_output(), mimetype='text/event-stream')
```

Demo

Starting the Flask app



The image shows a side-by-side comparison of two Visual Studio Code windows. The left window, titled 'AniGEN', displays a TypeScript file with imports for Firebase, GoogleGenerativeAI, and axios, and a function named 'Prompt'. The right window, titled 'flask-app', displays a Python Flask application with routes for a home page and a motions endpoint. Below the code editors, the terminal windows show the execution of the Flask application, displaying a warning about the development server and the URL http://127.0.0.1:5000.

```
import { db } from '../Firebase';
import { GoogleGenerativeAI, Part, HarmCategory, HarmBlockThr
import { getStorage, ref, listAll } from "firebase/storage";
import axios from 'axios';

function Prompt() {
  const inputRef = useRef<HTMLInputElement>(null);
  const storage = getStorage();
  const listRef = ref(storage, '0411mixamo/xbot/');

  const genAI = new GoogleGenerativeAI("AIzaSyBSTRnLPvp5AE

function motionExists(motions: any[], filesString: strin
```

```
from flask import Flask, Response
from flask_sse import sse
from flask import request, jsonify
import config as Config
from flask_cors import CORS

app = Flask(__name__)
CORS(app)
app.register_blueprint(sse, url_prefix='/stream')

@app.route('/')
def home():
    # Reset the config file
    return 'This is the AniGEN Flask app to execute anigen-b

# This method will receive a json which will contain names o
@app.route('/motions', methods=['POST'])
def motions_receive():
    data = request.json
    # Store motions in the config file
    Config.MOTIONS = data['motions']
```

```
(base) PS C:\Users\User\Desktop\FYP\AniGEN>
```

```
(anigen-venv) (base) PS C:\Users\User\Desktop\FYP\flask-app> flask run
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deploy
ment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

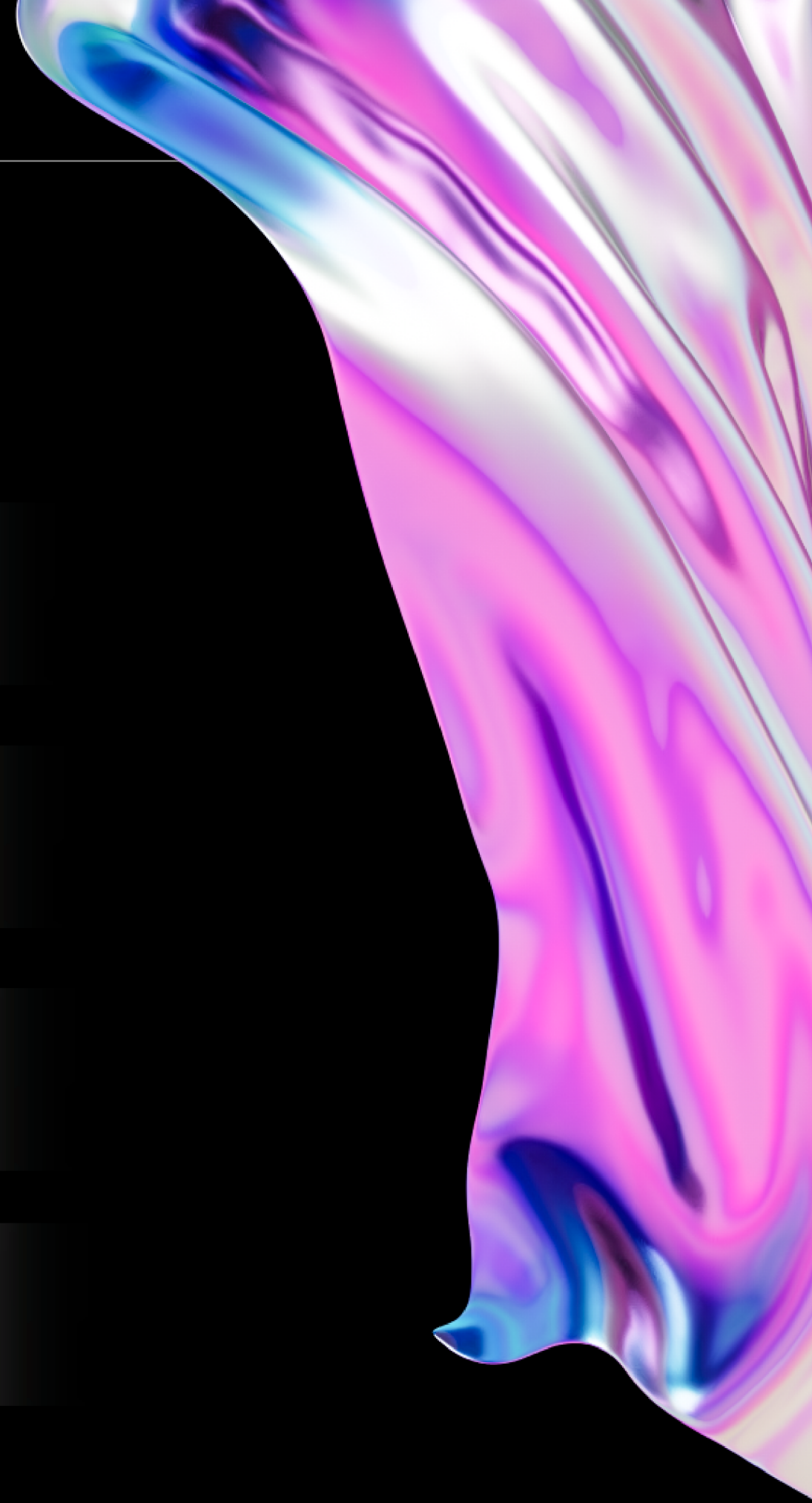

Difficulties Encountered

 [Redacted text]

 Keeping up with Project Schedule

 [Redacted text]

 [Redacted text]



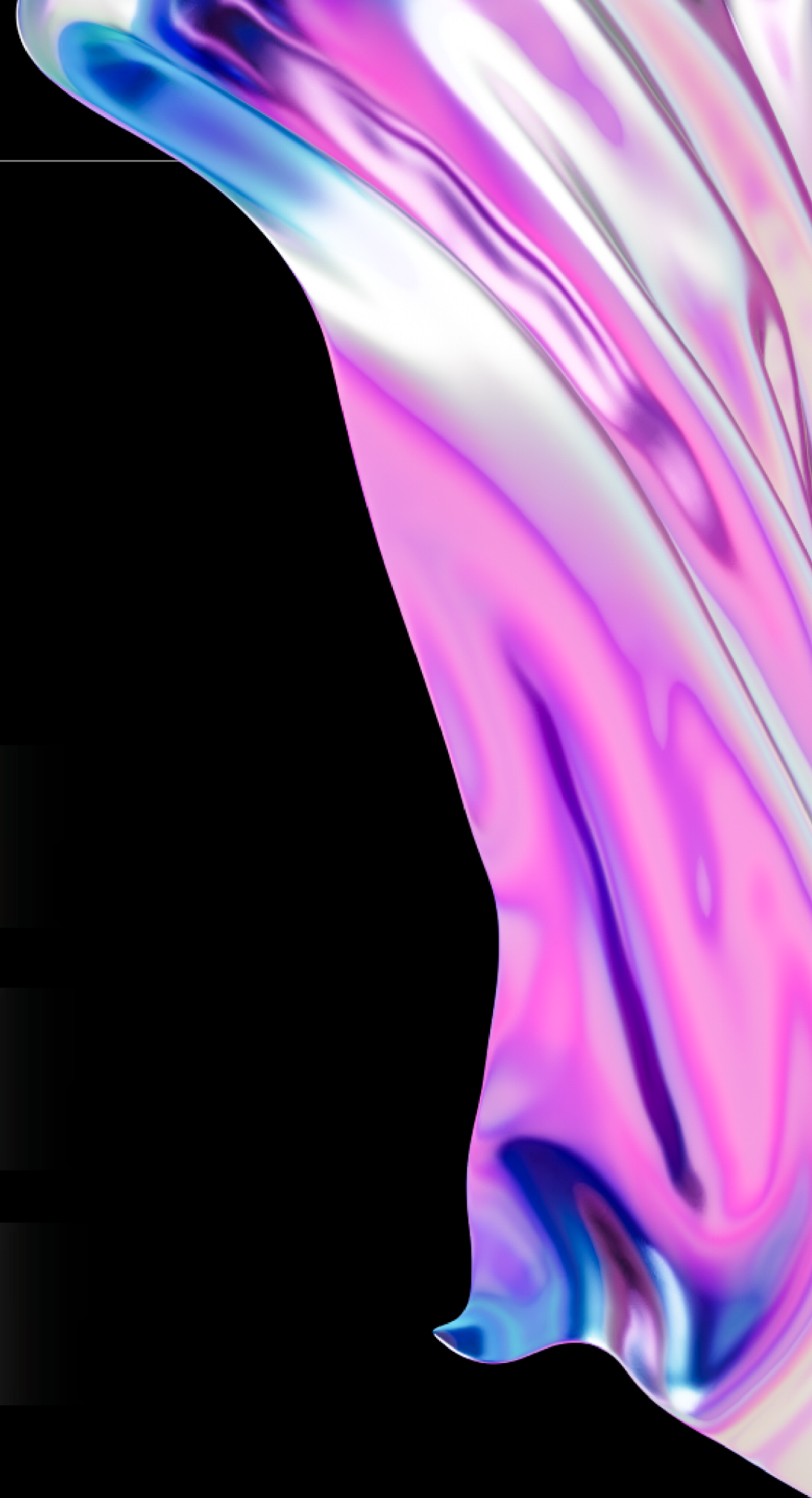
Difficulties Encountered

 Using blender in background for anigen-blender-utils

 Keeping up with Project Schedule







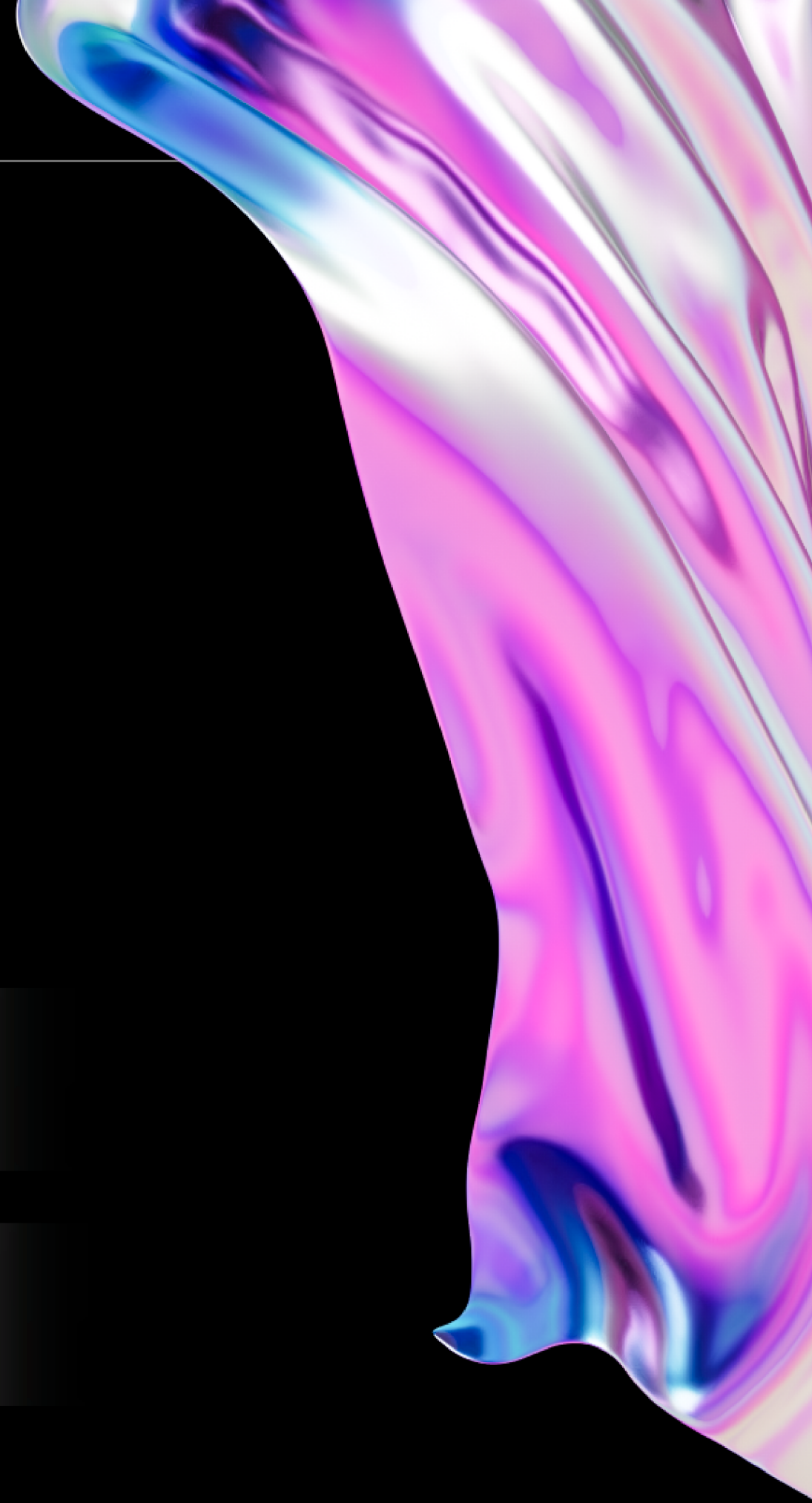
Difficulties Encountered

 Using blender in background for anigen-blender-utils





 Difficulty in working with non in-place motions

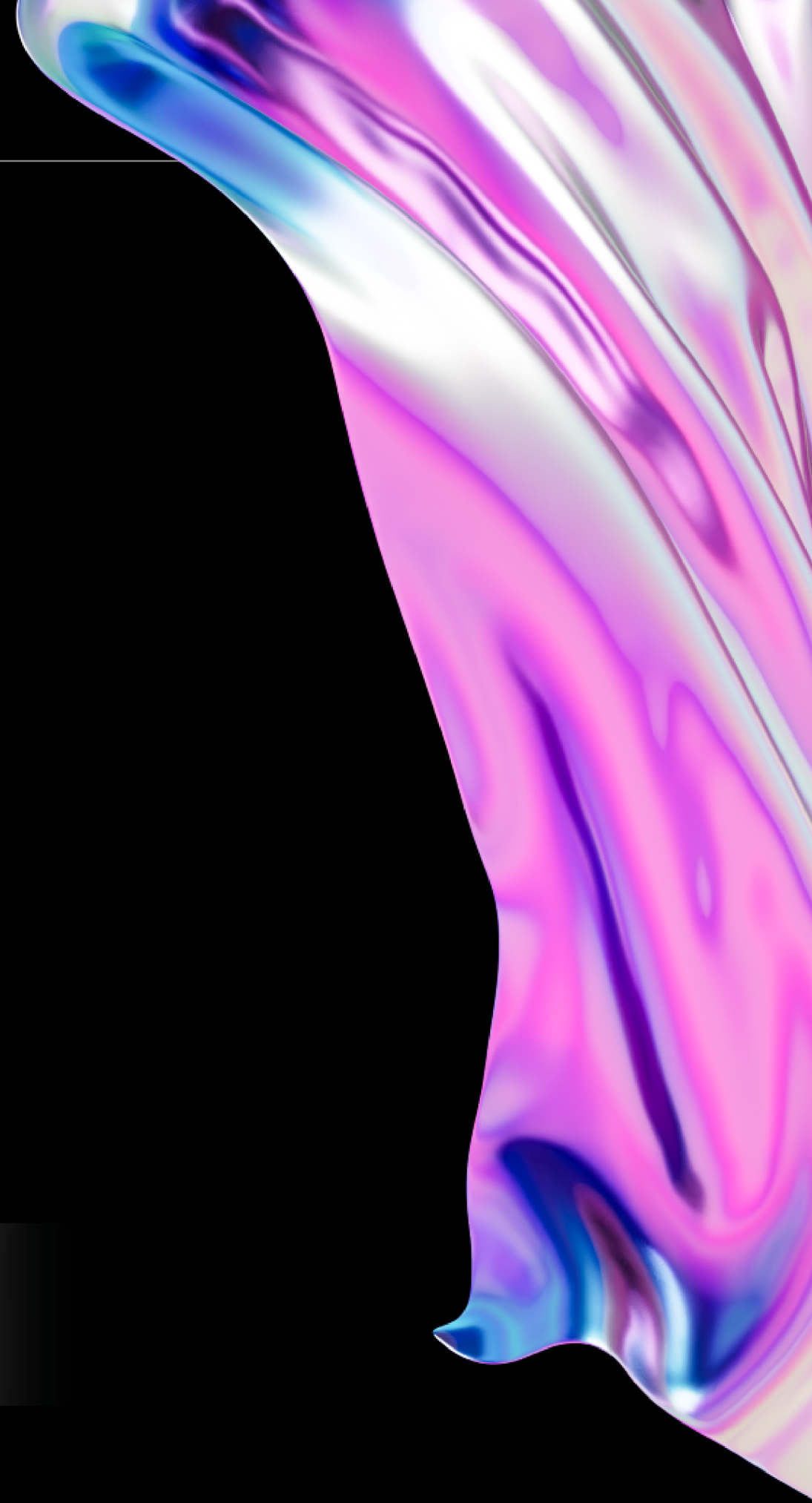










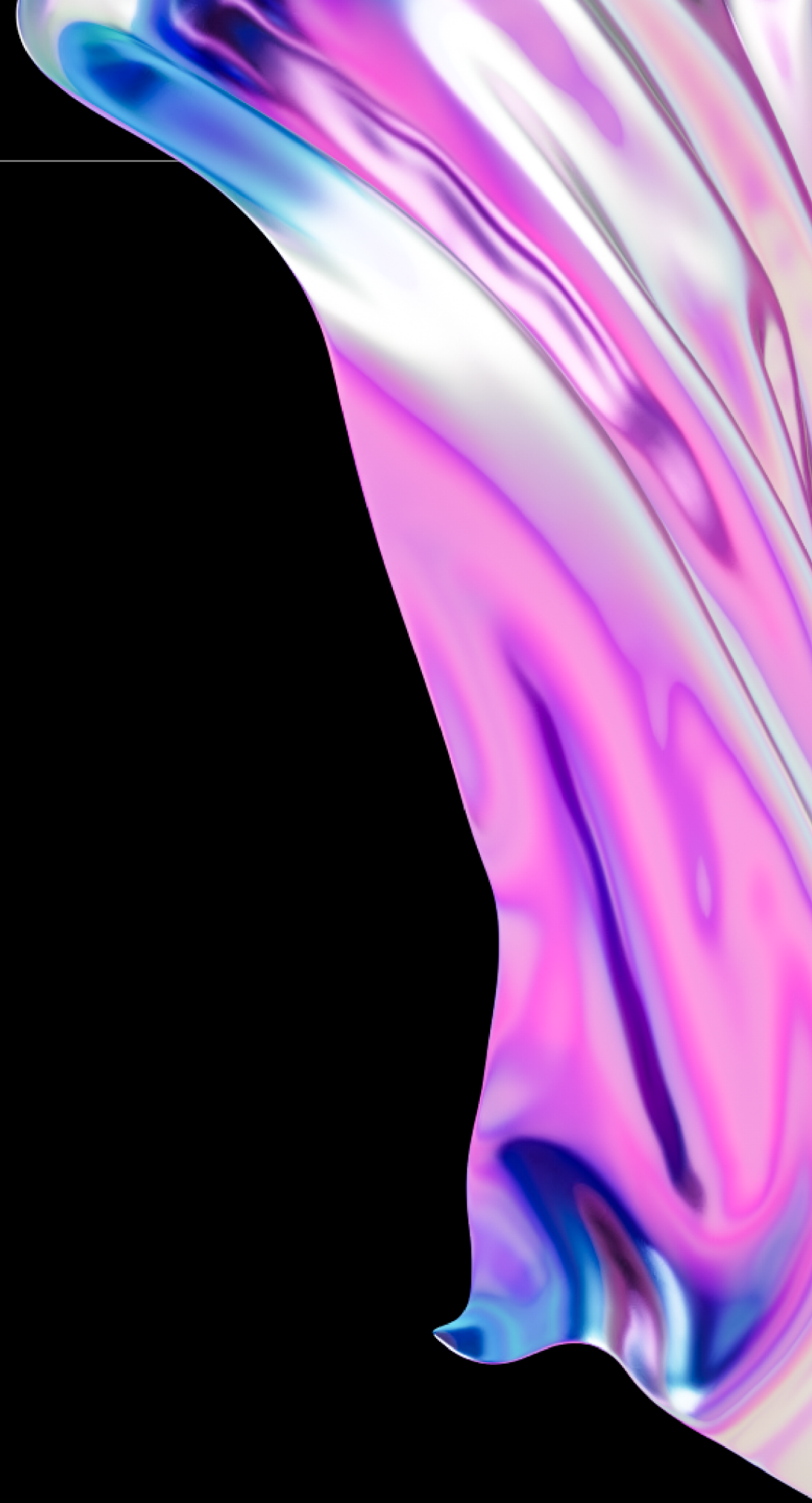
Difficulties Encountered

-  Using blender in background for anigen-blender-utils
-  Difficulty in working with non in-place motions
-  CORS issue during communication between webapp and flask
- 



Difficulties Encountered

-  Using blender in background for anigen-blender-utils
-  Difficulty in working with non in-place motions
-  CORS issue during communication between webapp and flask
-  LLM Implementation



Technical Limitation

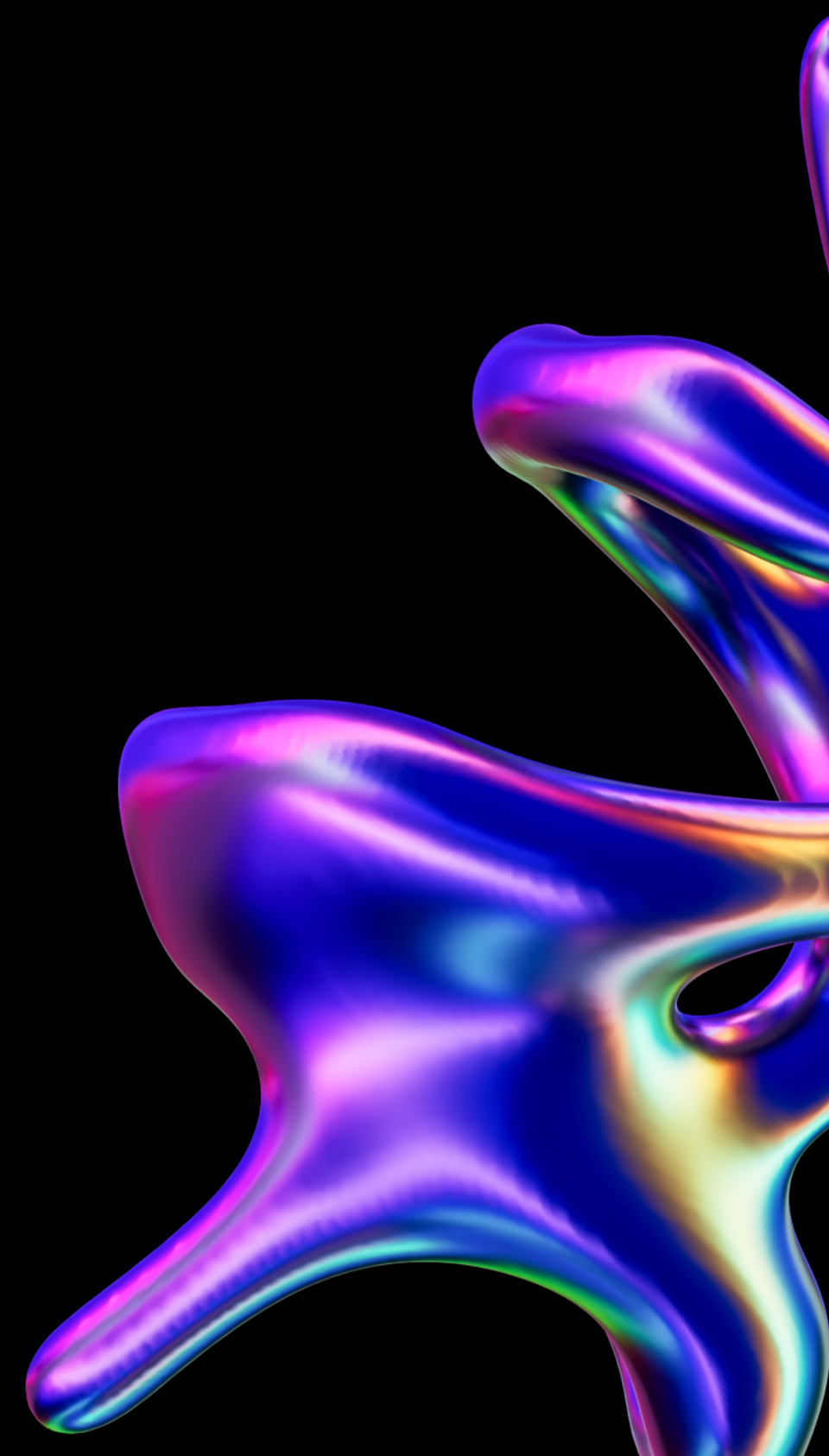
Why **Gemini** 1.0 Pro?

Price - API credits ^[19]

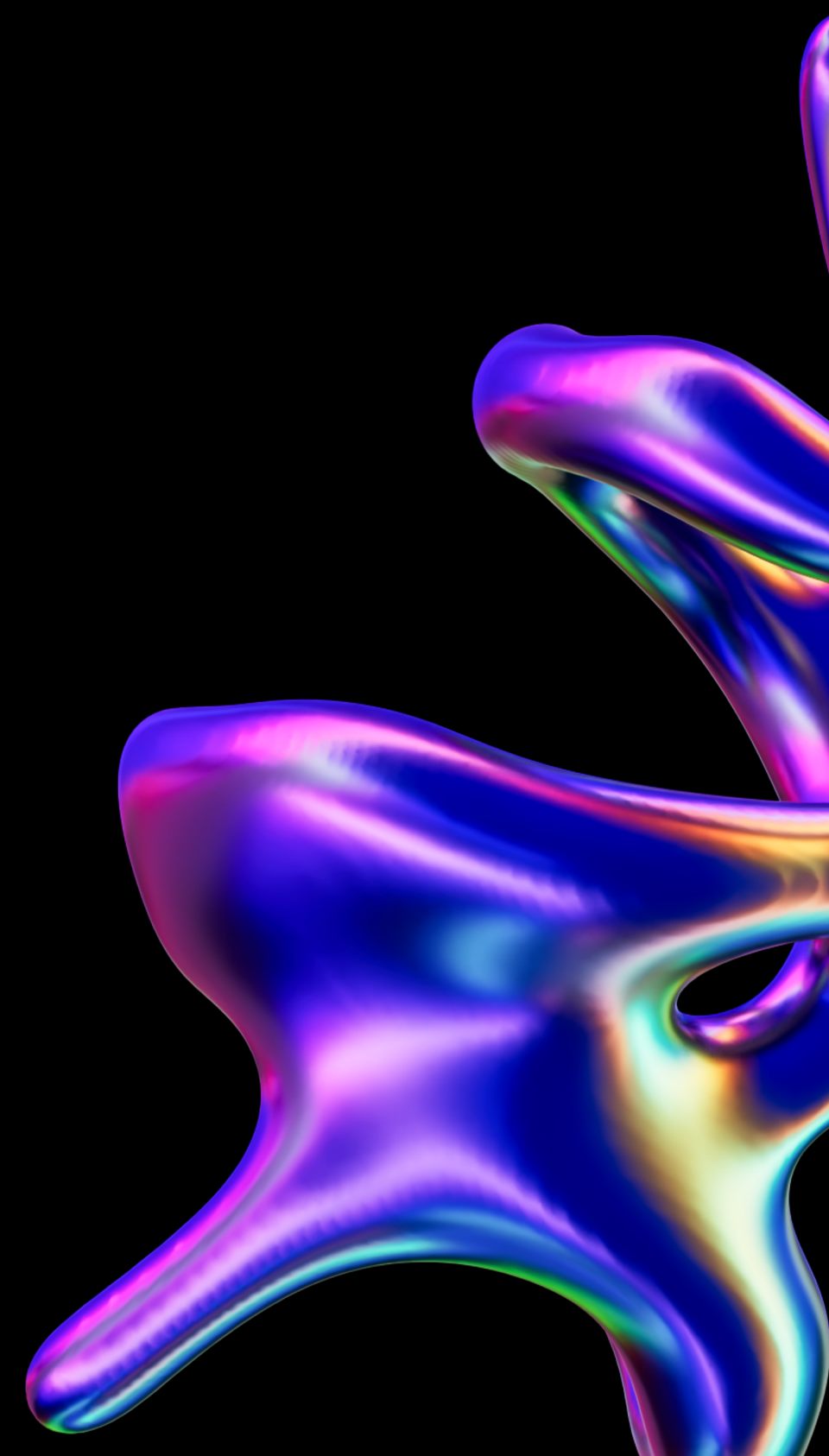
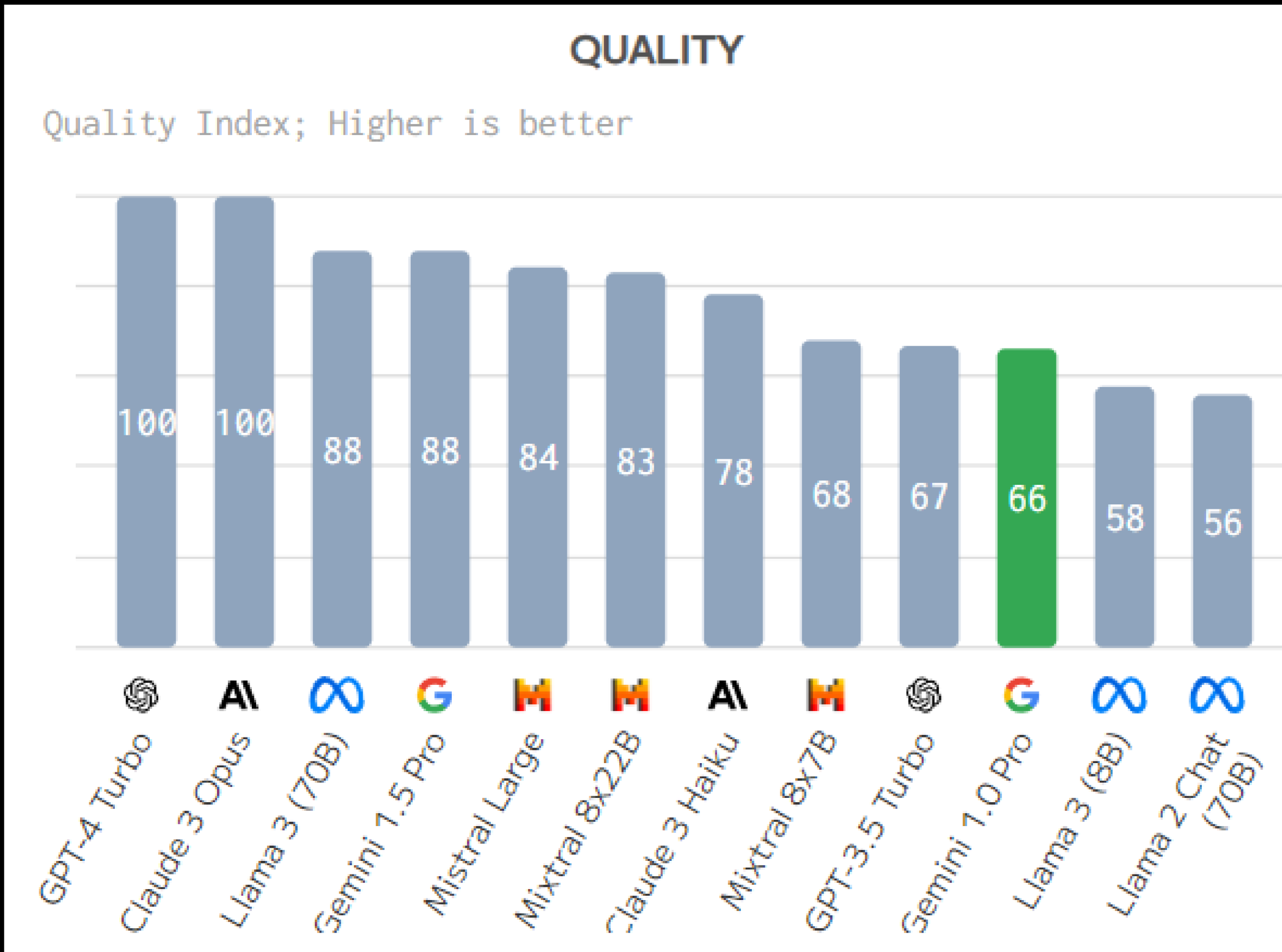
Data quality - Simple extraction

Speed - Cut down processing time

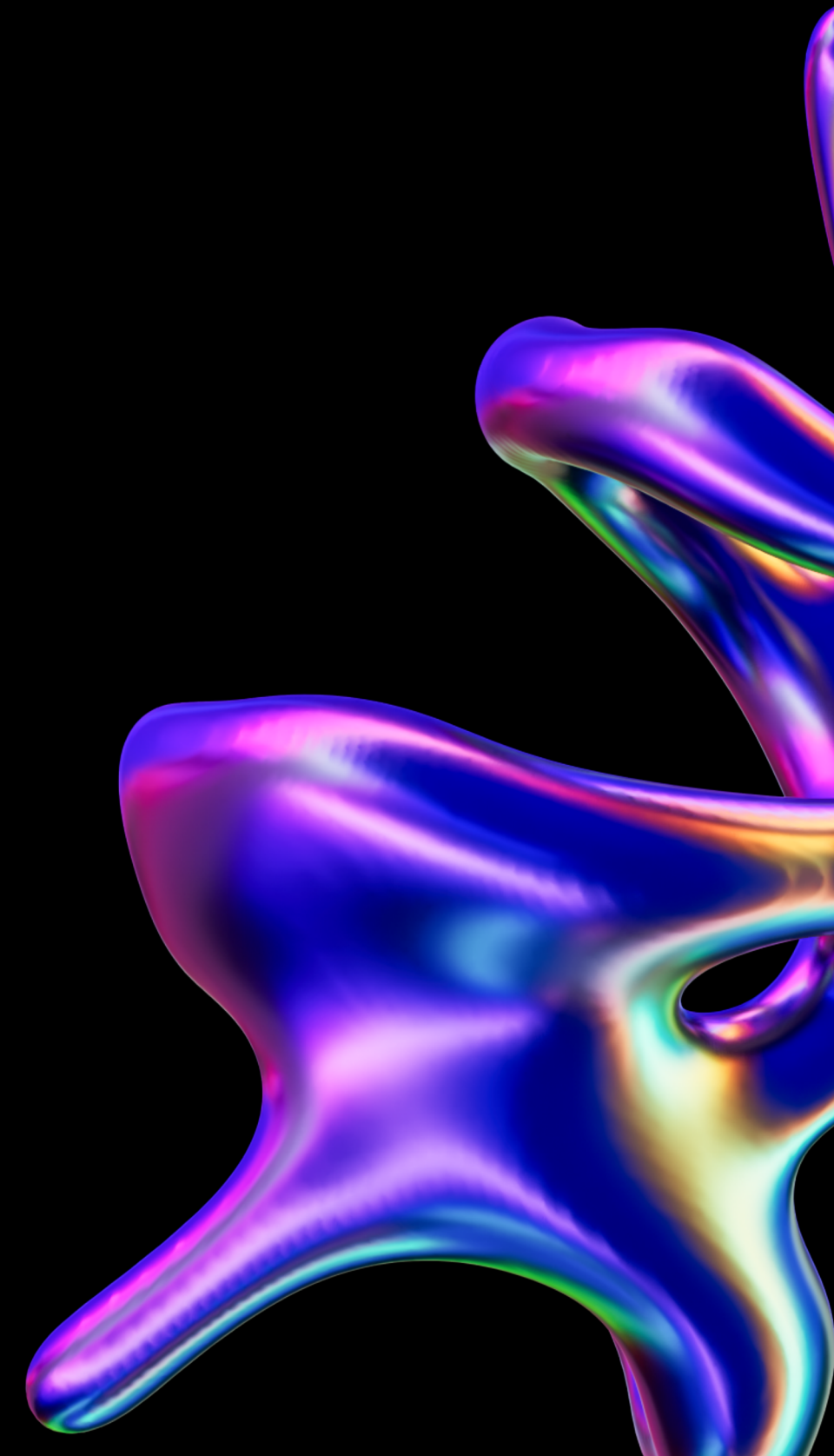
*Ranked against base tier LLMs



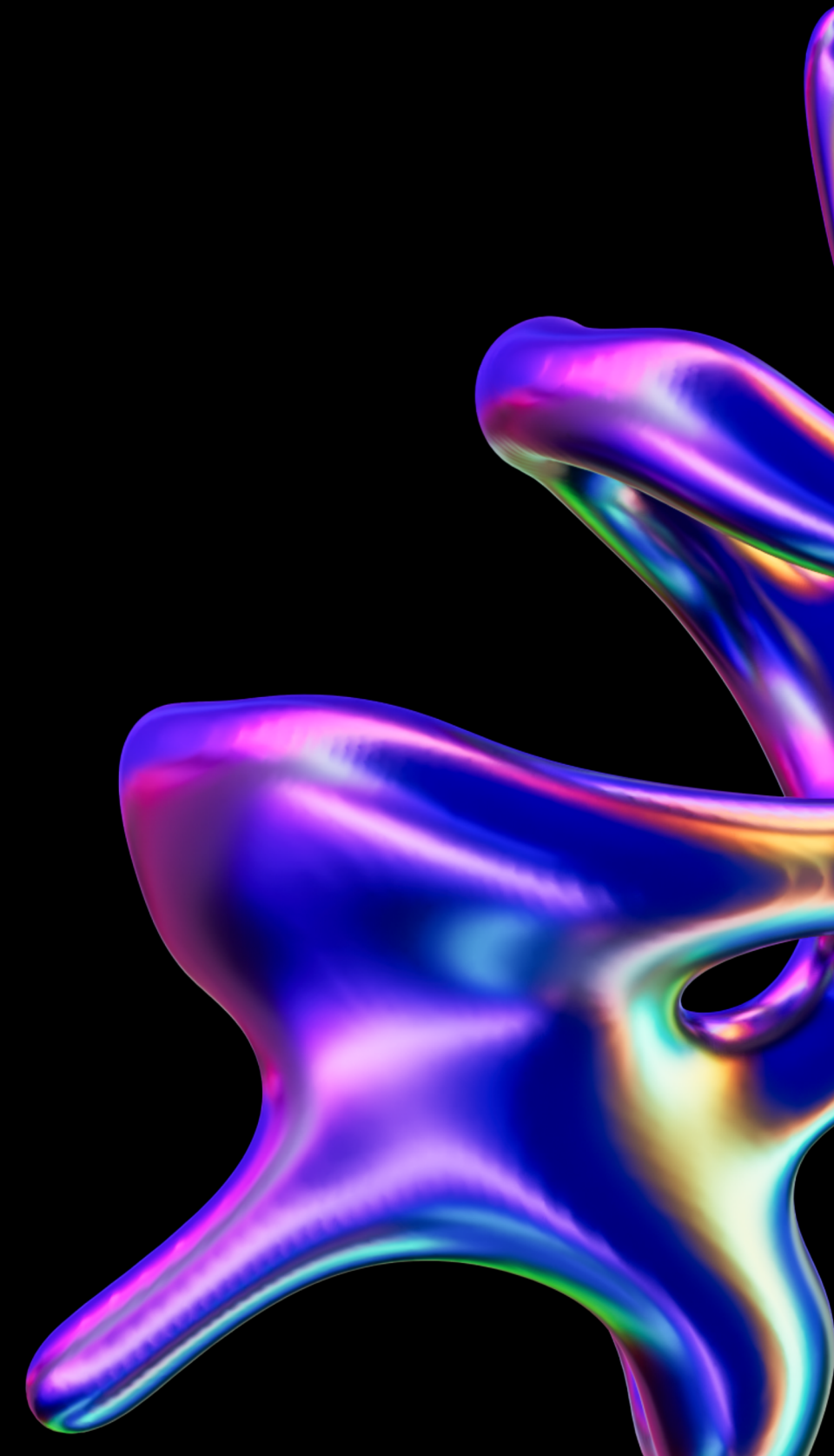
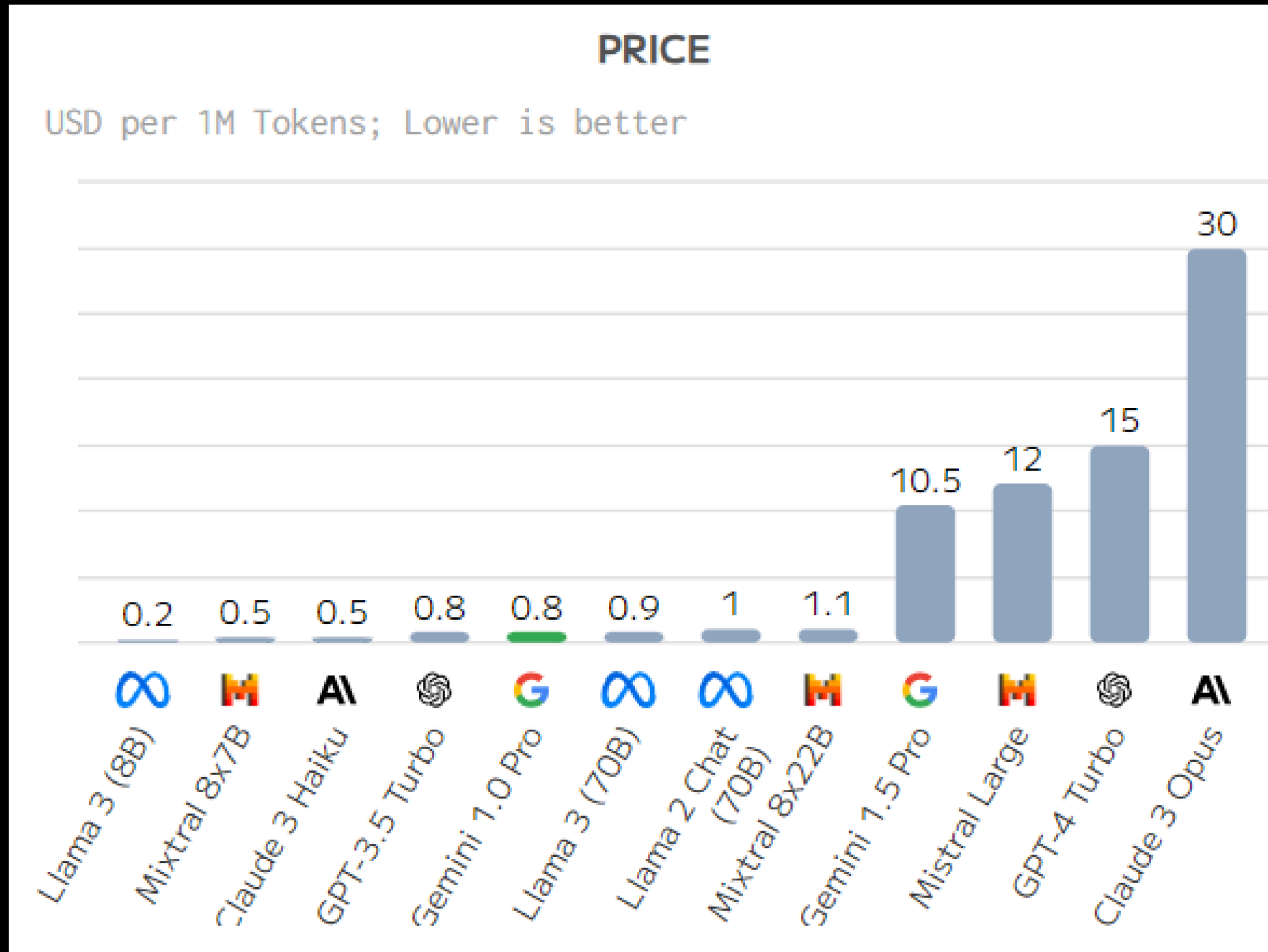
Technical Limitation



Technical Limitation



Technical Limitation



05. Conclusion

- 05.1 Conclusion
- 05.2 AniGEN beyond 2024



Conclusion

01

**A new solution
turning text into
avatar animation**

Aimed for bridging the skill gap and make 3D avatar animation more accessible for studios and creators.

02

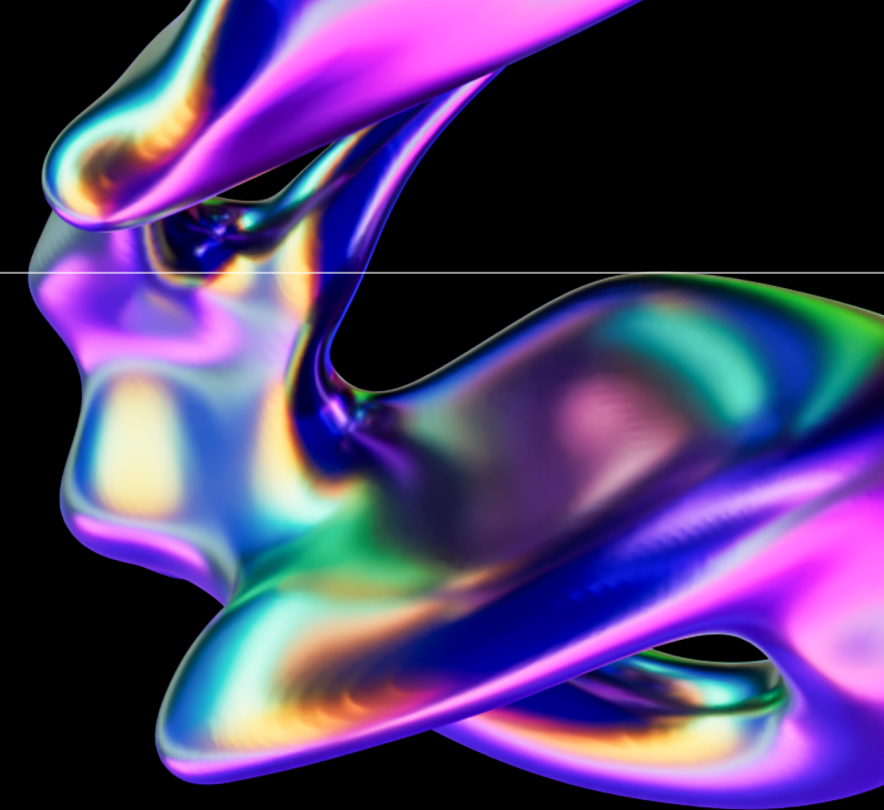
**Powered by React,
Python, Gemini and
Blender**

Transforming a text prompt into an interactive avatar animation video with the power of NLP and Blender..

03

**Project completion
achieved in April
2024**

Video, Poster and Website now available
Reports available on 26/4



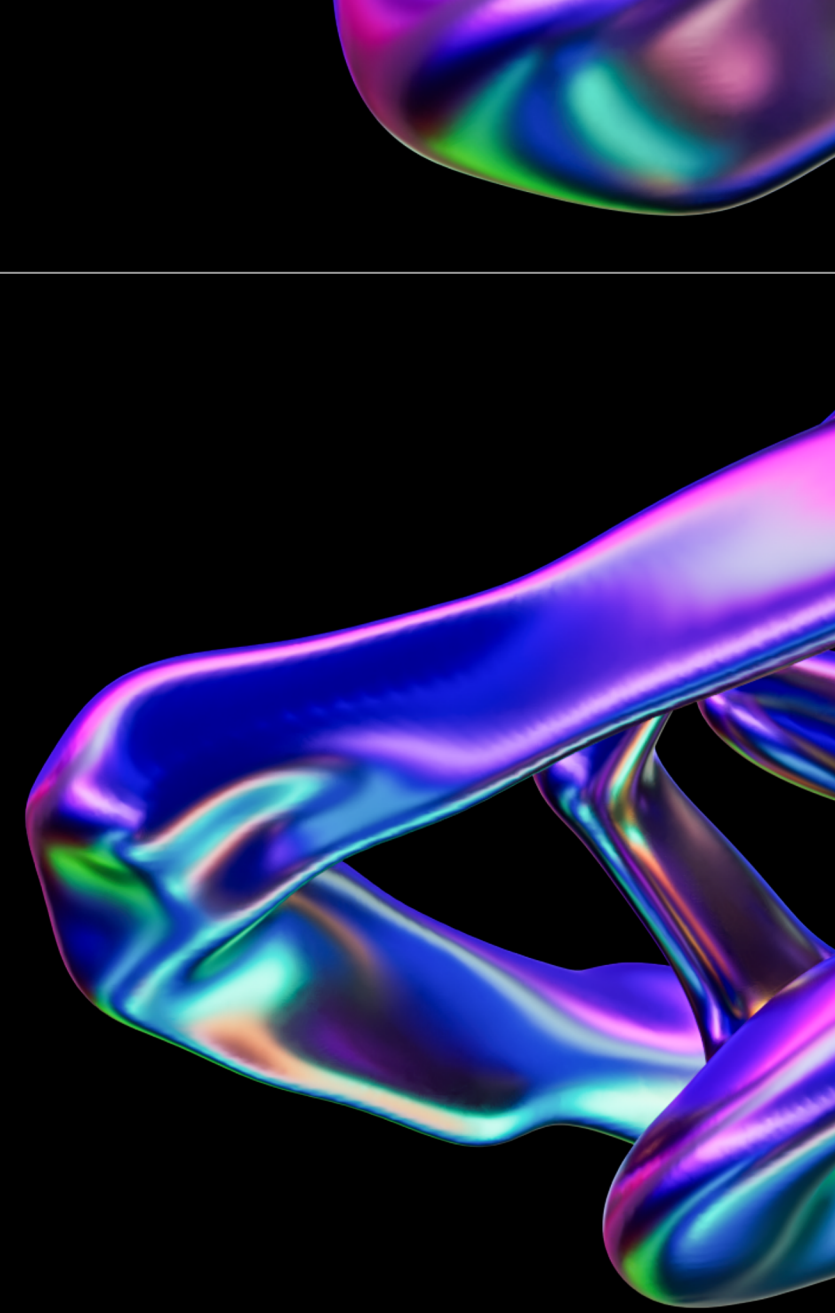
AniGEN beyond 2024

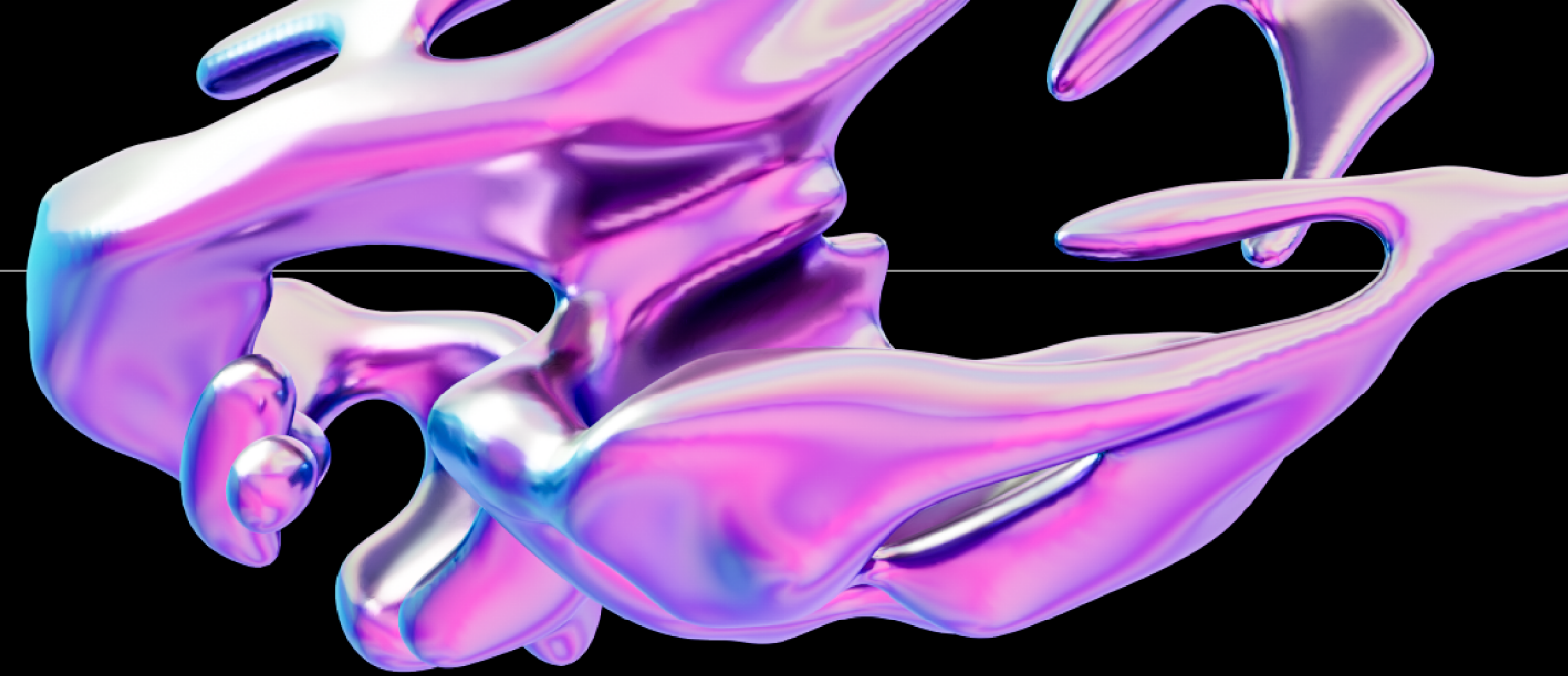
- Uploading custom 3D models
- Generating Dance Videos alongside music
- Blending non in-place motions



- Scaling database to online videos for motion data to include trending videos (gathering motion data from online videos)

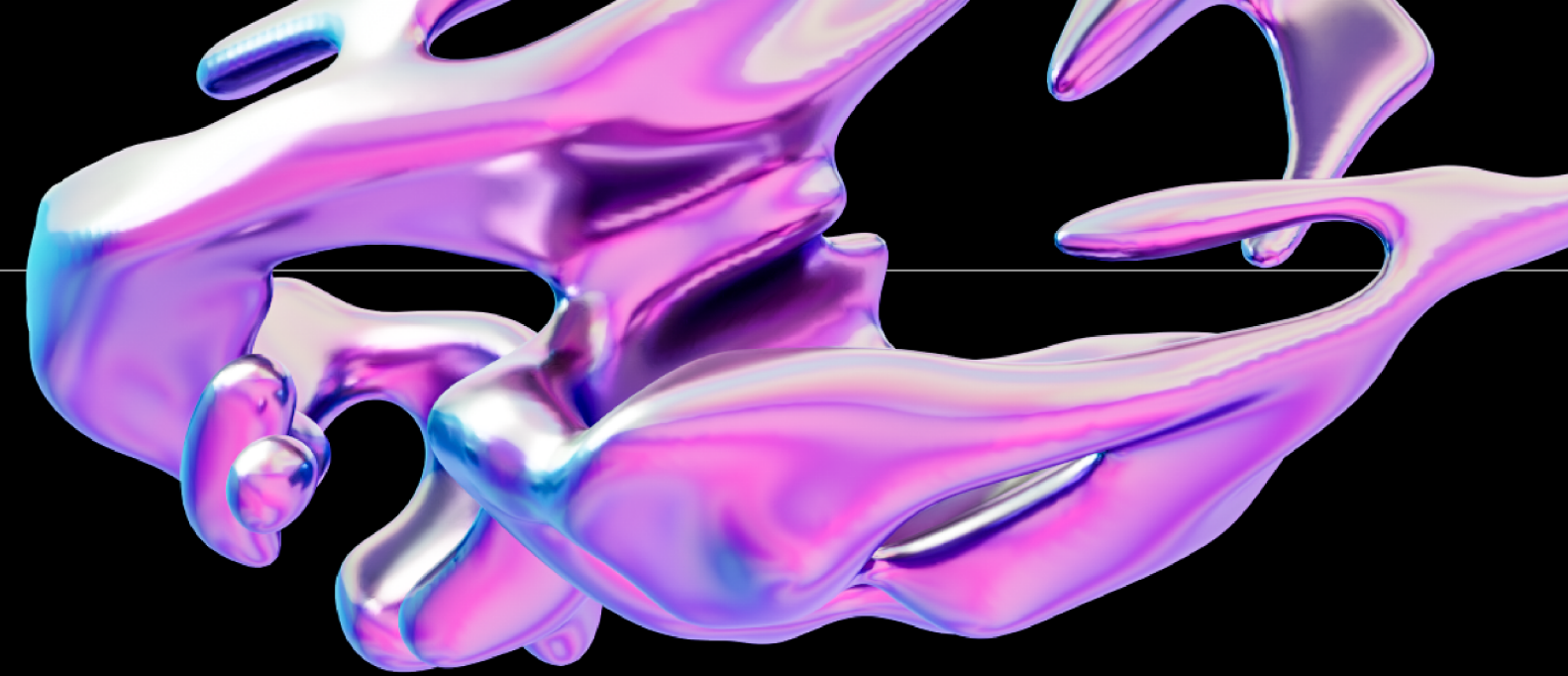
AniGEN beyond 2024





Reference List

- [1] "GTA 6 Price: Most Expensive Entertainment Product of All Time?," Gaming Bible, [press@ladbiblegroup.com, https://www.gamingbible.com/news/platform/gta-6-price-most-expensive-entertainment-product-all-time-387291-20230719](https://www.gamingbible.com/news/platform/gta-6-price-most-expensive-entertainment-product-all-time-387291-20230719) (accessed November 12, 2023).
- [2] "Most expensive film productions worldwide," Statista, <https://www.statista.com/statistics/273416/most-expensive-film-productions-worldwide/> (accessed November 12, 2023).
- [3] "Orc Face Motion Capture," Reddit, https://www.reddit.com/r/gifs/comments/1t1rx7/orc_face_motion_capture/ (accessed November 12, 2023).
- [4] "Tracking, Matchmove & Rotomation," Boss VFX Studio, <https://www.bossvfxstudio.com/tracking-matchmove-rotomation/> (accessed November 12, 2023).
- [5] DeepMotion. "Home - DeepMotion." <https://www.deepmotion.com/>. (accessed November 12, 2023).
- [6] Runway. "Make the Impossible," Runway. <https://runwayml.com/>. (accessed January 12, 2024).
- [7] R. D. Hernandez, "The Model View Controller Pattern – MVC Architecture and Frameworks Explained," freeCodeCamp, Apr. 19, 2021. Available: <https://www.freecodecamp.org/news/the-model-view-controller-pattern-mvc-architecture-and-frameworks-explained/>
- [8] React. "React - The Library for web and native user interfaces", Meta Open Source. <https://react.dev/>. (accessed April 24, 2024).
- [9] Vite. "Vite | next generation frontend tooling," Vite, 2019. <https://vitejs.dev/>. (accessed April 24, 2024).
- [10] Bootstrap. "Bootstrap · The most popular HTML, CSS, and JS library in the world.", Bootstrap Team. <https://getbootstrap.com/>. (accessed April 24, 2024)



Reference List

- [11] Tailwind CSS. "Tailwind CSS - Rapidly build modern websites without ever leaving your HTML." Tailwind Labs. <https://tailwindcss.com/>. (accessed April 23, 2024).
- [12] Gemini. "Gemini - Chat to supercharge your ideas". Google LLC. <https://gemini.google.com/>. (accessed April 23, 2024).
- [13] Blender Authors, "Quickstart - Blender Python API," docs.blender.org. Available: https://docs.blender.org/api/current/info_quickstart.html. (accessed April 23, 2024).
- [14] Mixamo. "Get animated." Adobe Systems Incorporated. <https://www.mixamo.com/>. (accessed April 23, 2024).
- [15] Python. "Welcome to Python.org." Python Software Foundation. <https://www.python.org/>. (accessed April 23, 2024).
- [16] Firebase. "Firebase | Google's Mobile and Web App Development Platform". <https://firebase.google.com/>. (accessed April 23, 2024).
- [17] Flask. "Flask | The Pallets Projects". The Pallets Projects. <https://palletsprojects.com/p/flask/>. (accessed April 23, 2024).
- [18] Blender Authors, "Nonlinear Animation - Blender 4.1 Manual," docs.blender.org. Available: <https://docs.blender.org/manual/en/latest/editors/nla/index.html>. (accessed April 23, 2024).
- [19] Google, "Gemini API Pricing," Google AI for Developers. Available: <https://ai.google.dev/pricing>. (accessed April 23, 2024).
- [20] Artificial Analysis, "Gemini Pro Providers," Artificial Analysis AI, Available: <https://artificialanalysis.ai/models/gemini-pro/providers>. (accessed April 23, 2024).