“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

[1]
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

Advanced equipment & technical knowledge required

Big Companies
Background

Independent Artists

[4] Rotomation

Issues: Time-consuming
Background

Independent Artists

Issues: Inconsistent and jittering motion

[5] Deepmotion - Video motion tracking
Motivation

Text-to-video generation Model

Text Prompt → Text-to-video Generation → Non-interactive Video output
Motivation

Text-to-video generation Model

Text Prompt → Text-to-video Generation → Non-interactive Video output
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

1. Input: Text Prompt
2. Relevant motion data recognised using NLP and queried
3. More than one motion?
4. Rendering the motion(s)
5. No
The Input-Process-Output workflow of AniGEN
The Input-Process-Output workflow of AniGEN

- **Input**: Text Prompt
- **Frontend**
  - Relevant motion data recognised using NLP and queried
  - More than one motion?
    - Yes: Blending sequences of motions
    - No: Rendering the motion(s)
- **Backend**
- **Frontend**
  - Output: Video and FBX file
How AniGEN implements the Model-View-Controller architecture

- **anigen-blender-utils**
  - (Model, i.e. the motion-blending logic)

- **anigen-flask-app**
  - (Controller / Abstraction Layer, i.e. the back-end)

- **AniGEN web app**
  - (View / User Interface, i.e. the front-end)

Data and Execution
Command invoking

Editable .fbx file + Exported video file

GET & POST requests to send list of motion file names, updating which character to select, execute blender script & get status on rendering & exporting
Tech Stack

- **AniGEN webapp**
    - (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[16]
    - (Back-end code)
  - FireBase[17]
    - (Database)
  - Flask[18]
    - (Python Micro web framework)
  - Postman[19]
    - (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

![Blend settings](image-url)
Blender Utils
Final Results

Implementing
  • Motion Blending of In-Place Animation

Final Output:
Implementing  
• Motion Blending of In-Place Animation
Blender Utils
Final Results

```
def main():
    import the blender file and the fbx files
    import_path = Config.IMPORT_PATH
    motions = Config.MOTIONS
    fbx_files = [f"motion{num}.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
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

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

```python
# This method will receive a json which will contain the information about the blender character
@app.route('/character', methods=['POST'])
def character_receive):
    # Extract the character name from the json
    data = request.json
    character = data['character']
    # Store character information in the config file's BLEN_PATH
    blend_path = r'C:\Users\User\Desktop\FYP\flask-app\{}.blend'.format(character)
    Config.BLEN_PATH = blend_path
    write_config_file()
    return '', 200
```
Flask App Final Results

Endpoint at:

```
@app.route('/notification', methods=['GET'])
def notification():
    # Extract the code and status from the config file
    payload_cs = jsonify({'code': Config.CODE, 'status': Config.STATUS})
    payload_cs.headers['Access-Control-Allow-Origin'] = '*'
    return payload_cs, 200
```

Implemented

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

'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
Starting the Flask app
Difficulties Encountered

- Keeping up with Project Schedule
Difficulties Encountered

- Using blender in background for anigen-blender-utils
- Keeping up with Project Schedule
- [Blank]
- [Blank]
- [Blank]
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

Data quality - Simple extraction

Speed - Cut down processing time

*Ranked against base tier LLMs*
Technical Limitation

Quality Index; Higher is better

<table>
<thead>
<tr>
<th>Model</th>
<th>Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPT-4 Turbo</td>
<td>100</td>
</tr>
<tr>
<td>Claude 3 Opus</td>
<td>100</td>
</tr>
<tr>
<td>Lilana 3 (70B)</td>
<td>88</td>
</tr>
<tr>
<td>Gemini 1.5 Pro</td>
<td>88</td>
</tr>
<tr>
<td>Mistral Large</td>
<td>84</td>
</tr>
<tr>
<td>Mixtral 8x228</td>
<td>83</td>
</tr>
<tr>
<td>Claude 3 Haiku</td>
<td>78</td>
</tr>
<tr>
<td>Mixtral 8x78</td>
<td>68</td>
</tr>
<tr>
<td>GPT-3.5 Turbo</td>
<td>67</td>
</tr>
<tr>
<td>Gemini 1.0 Pro</td>
<td>66</td>
</tr>
<tr>
<td>Lilana 3 (8B)</td>
<td>58</td>
</tr>
<tr>
<td>Lilana 2 Chat (70B)</td>
<td>56</td>
</tr>
</tbody>
</table>
Technical Limitation

SPEED

Throughput in Tokens per Second; Higher is better

<table>
<thead>
<tr>
<th>Model</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>Llama 3 (78B)</td>
<td>197</td>
</tr>
<tr>
<td>Mixtral 8x78</td>
<td>96</td>
</tr>
<tr>
<td>Claude 3 Haiku</td>
<td>88</td>
</tr>
<tr>
<td>Semini 1.0 Pro</td>
<td>81</td>
</tr>
<tr>
<td>Llama 3 (70B)</td>
<td>74</td>
</tr>
<tr>
<td>Mixtral 8x228</td>
<td>59</td>
</tr>
<tr>
<td>GPT-3.5 Turbo</td>
<td>56</td>
</tr>
<tr>
<td>Llama 2 Chat (70B)</td>
<td>48</td>
</tr>
<tr>
<td>Semini 1.5 Pro</td>
<td>45</td>
</tr>
<tr>
<td>Mistral Large</td>
<td>27</td>
</tr>
<tr>
<td>Claude 3 Opus</td>
<td>26</td>
</tr>
<tr>
<td>GPT-4 Turbo</td>
<td>17</td>
</tr>
</tbody>
</table>
Technical Limitation

USD per 1M Tokens; Lower is better
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

Reference List