Project: Constraint-Based Creative Generation with AI

Context:
This project explores the relationship between generative AI and formal constraints through the task of dance choreography generation. This assignment is grounded in research on computational creativity, constraint-based generation, and human–AI co-creation.

In many creative domains, constraints play a generative role: they limit the solution space while simultaneously shaping form and structure. By separating expressive generation (natural language) from formal validation (rule-based checking), this project makes constraints explicit and inspectable.

The iterative generate–validate–refine loop reflects established models of creative cognition, where artifacts are progressively shaped by feedback rather than produced in a single step. Rather than evaluating creativity as free-form novelty, the project focuses on structured creative generation under explicit spatial and temporal rules.
The final aim will be the design of a pipeline that generates choreographies in natural language, converts them into a formal representation, and validates them using a rule-based system.

Task Description

You can find here https://www.dropbox.com/scl/fo/e7txid7g5kfdy98ssy112/AAsfX4OOef-EHHp5UJSuqgY?rlkey=y91pq7v8anai5ugg4v25ybzxv&st=1q1fp0n3&dl=0 the following material:

1) a prompt to generate dance choreographies into structured JSON,

2) a validator (in Python) that validates spatial, temporal, and physical constraints.

Your task is to generate multiple valid choreographies and analyze how creativity emerges within the imposed constraints.
This part should be conducted by choosing at least two different LLMs. The prompt must be used via API and code for each LLM tested. Please, do not use the graphical interface to do that.

Required Steps
1. Choreography Generation (Natural Language)

Start by using the provided prompt to generate dance choreographies (60–90 seconds) involving 2–5 dancers. We ask you to analyse the prompt and eventually to improve it by studying some related works in literature.

The choreography must describe:

-entrances and exits,
-dancer movements and interactions,
-approximate timing,
-spatial locations on stage.

2. Formal Encoding (JSON Representation)

Each choreography must be generated as a structured JSON format.

The JSON must:

-map the stage onto a 25×25 grid (at least),
-represent movements as time-bounded events,

-encode spatial transitions explicitly,
-remain fully machine-interpretable.

3. Validation

Run each JSON choreography through the provided validator.

The validator enforces:

-spatial boundaries,
-collision avoidance,
-movement speed limits,
-temporal coherence,
-reasonable dancer activity.

Only choreographies that pass validation are considered physically plausible.

4. Iterative Refinement

If validation fails, revise the generation process.

This must involve:

-modifying prompts,
-adjusting temporal or spatial constraints,
-regenerating choreographies.

Repeat the process until valid choreographies are obtained. You must save each process and each number of iteration required to have a valid choreography and for each LLM tested.

5. Comparative Analysis

Generate multiple distinct valid choreographies with different LLMs.

Analyze and discuss:

-similarities and differences in spatial organization,
-emergent patterns or motifs,
-how constraints influence stylistic variation.

The student must submit:

A repository with the code, the set of valid choreography JSON files, the info related to the number of vlid choreography and/or iteration required, a written report (3-5 pages).

Material:
https://www.promptingguide.ai/

Options for experiments with LLMs for Free

You can choose between the following main ways to access LLMs for text generation:

1- OpenAI GPT-3.5 Turbo (Free Trial)

GPT-3.5 Turbo is a version of OpenAI's language model for text generation.
New users sometimes get $5 of free credit upon creating an OpenAI account. This can be used to call GPT-3.5 via API.

Credits expire in a few months; once credits are used up, you need a paid plan.
Use the OpenAI Playground for browser experiments or OpenAI API for coding projects.
OpenAI API Pricing & Info: https://openai.com/api/pricing
OpenAI Playground: https://platform.openai.com/playground

2- Google Gemini via Google AI Studio (Free Tier)

Google's Gemini models (text, code, image, or multimodal generation).
Google AI Studio provides a free tier for many Gemini models (Flash, Lite).
Free tier has daily input/output limits. Some larger models require billing.
Sign up at Google AI Studio, create an API key, and start calling Gemini models.
Google AI Studio: https://aistudio.google.com/welcome
Google Gemini API Pricing & Free Tier: https://ai.google.dev/pricing

3- Hugging Face (Hosted Models via API)

Hugging Face hosts many open-source LLMs that can be used via API without downloading models.
Free tier allows limited API calls per day. You can experiment with many models, including GPT-2, MPT, LLaMA derivatives, and others.
Get a free Hugging Face account, generate an API token, and make POST requests to hosted models.
Hugging Face Model Hub: https://huggingface.co/models
Hugging Face Inference API Guide: https://huggingface.co/docs/api-inference