Procedural Content Generation Mini Project

In this project, you will explore procedural content generation (PCG) by implementing a system that generates game levels. You'll use deep learning techniques to create, analyze, and generate novel game layouts.

Environment Selection

While `Sokoban` from `Jumanji`[1] is recommended, you're encouraged to explore and choose any environment that meets these criteria:

- Levels can be procedurally generated
- Different object types are represented by distinct channels in the input

Spend some time researching and playing your chosen game to understand its rules and mechanics thoroughly.

Core Tasks

1. Dataset Creation:
   - Generate a substantial dataset of levels (aim for at least 1000)
   - Represent levels as 3D tensors (height x width x channels)
2. Autoencoder Implementation:
   - Design and train an autoencoder to compress and decompress levels.
   - Use `optax`[2] for optimization for your model weights.
   - Experiment with different architectures and latent space dimensions
3. Level Generation and Analysis:
   - Generate new levels by sampling from the latent space
   - Evaluate the quality and playability of generated levels
4. Advanced Exploration:
   - Implement a variational autoencoder (VAE) and compare results
   - Set the latent space to 2D and visualize it
   - Analyze nearest neighbors of specific levels in the latent space

Suggested Extensions

- Implement multiple PCG algorithms and compare their outputs
- Create a simple AI to play your generated levels

Deliverables

1. Code: A well-commented `.py` file containing your implementation
2. Report (300 words max):
   - Description of your approach and key decisions
   - Analysis of results and generated levels
   - Discussion of challenges faced and potential improvements
   - Reflection on what you learned about PCG and deep learning
3. Visualization: An image or interactive visualization of your latent space

Deadline

September 29th, 23:59 on LearnIT. Good luck, have fun, talk to each other, and talk to me.

---

[1] https://instadeepai.github.io/jumanji/environments/sokoban/

[2] https://optax.readthedocs.io/en/latest/