



Causal Inference in Procedural Dungeon Generation: DunGen

Alejandro Zapata Acosta and Andrew Forney

<https://github.com/CapnSquirrel/DungeonPCG>

Introduction

Procedural content generation (PCG) speeds up development time in games. This project explores Bayesian Network (BN) causal inference as a PCG method for random dungeon generation. Role-playing game (RPG) dungeons are classically known for their labyrinthian layout and archetypal room types.

DunGen is an application which takes as input the user's desired dungeon size (s, m, l) and room types (e.g. treasure, trap, empty, etc.), and produces a 2D RPG dungeon layout in the form of a graph.

Methods

We model a dungeon's structure with a directed acyclical graph (DAG) and use it as the structure for the Causal BN (CBN). Once we have a CBN, it is used to randomly generate a dungeon via traditional Bayesian inference. Users can then keep parts of the generated dungeon that they like, and regenerate those they don't via use of counterfactual inference.

Cascading Inference

Cascading inference is how we refer to the process of performing inference as we traverse the CBN:

$simulation_steps = topological_sort(network_nodes)$

for step in simulation_steps:

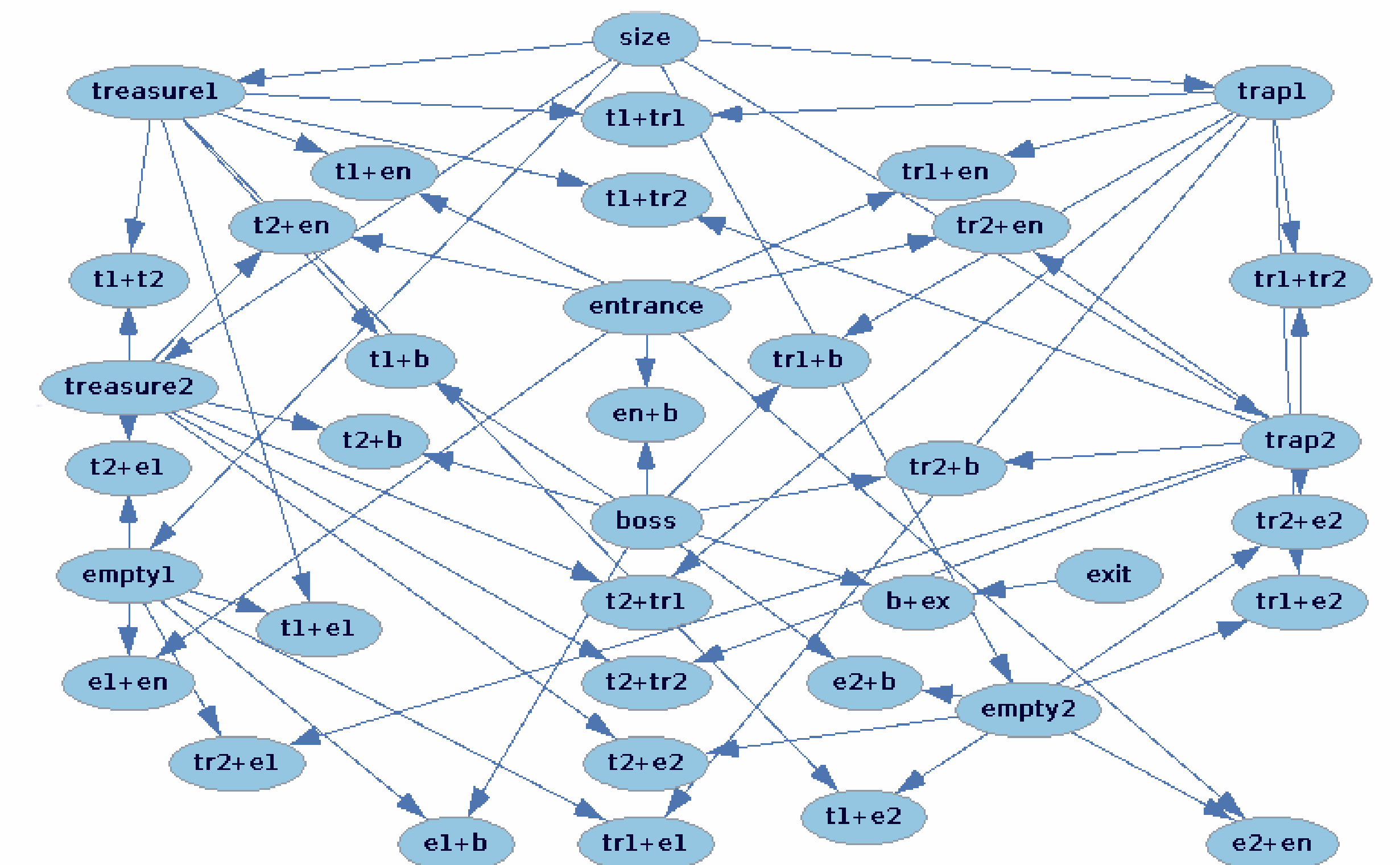
for node in step:

$node.parameters = P(room_exists | observed_rooms)$

write(network_nodes)

Inference powered by **pomegranate**

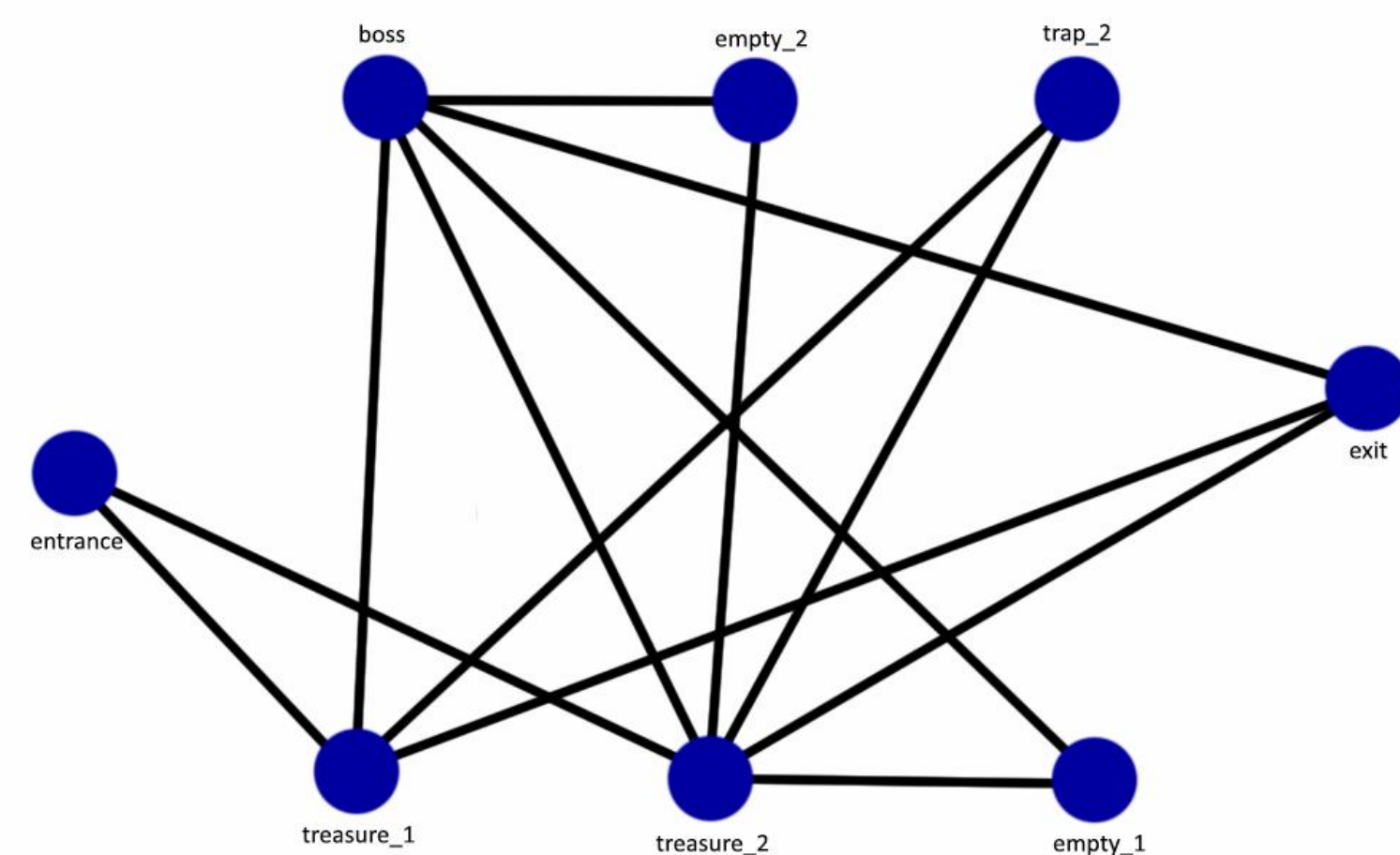
Sample Dungeon BN Structure



Process Mockup

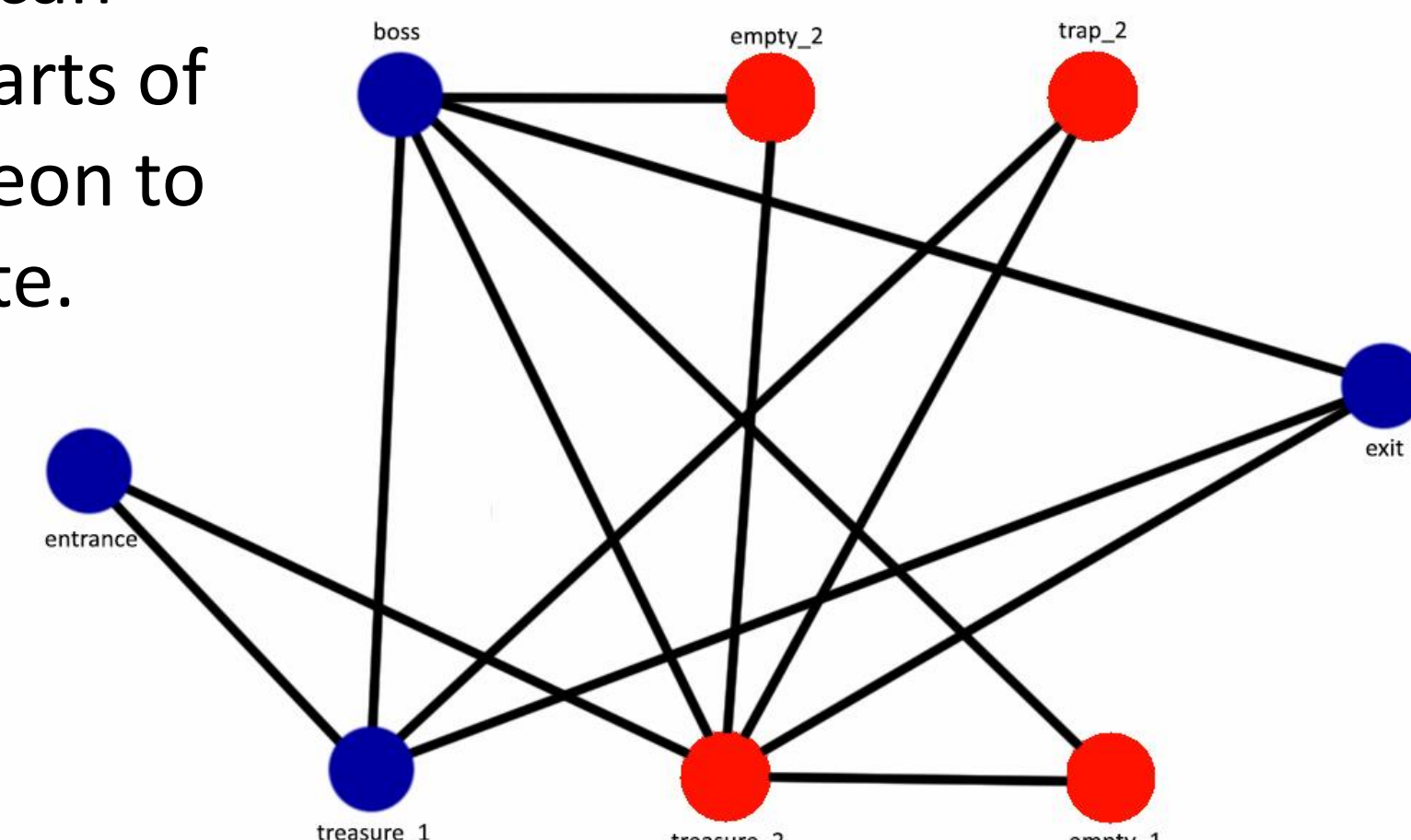
First output:

DunGen renders a randomly generated dungeon.



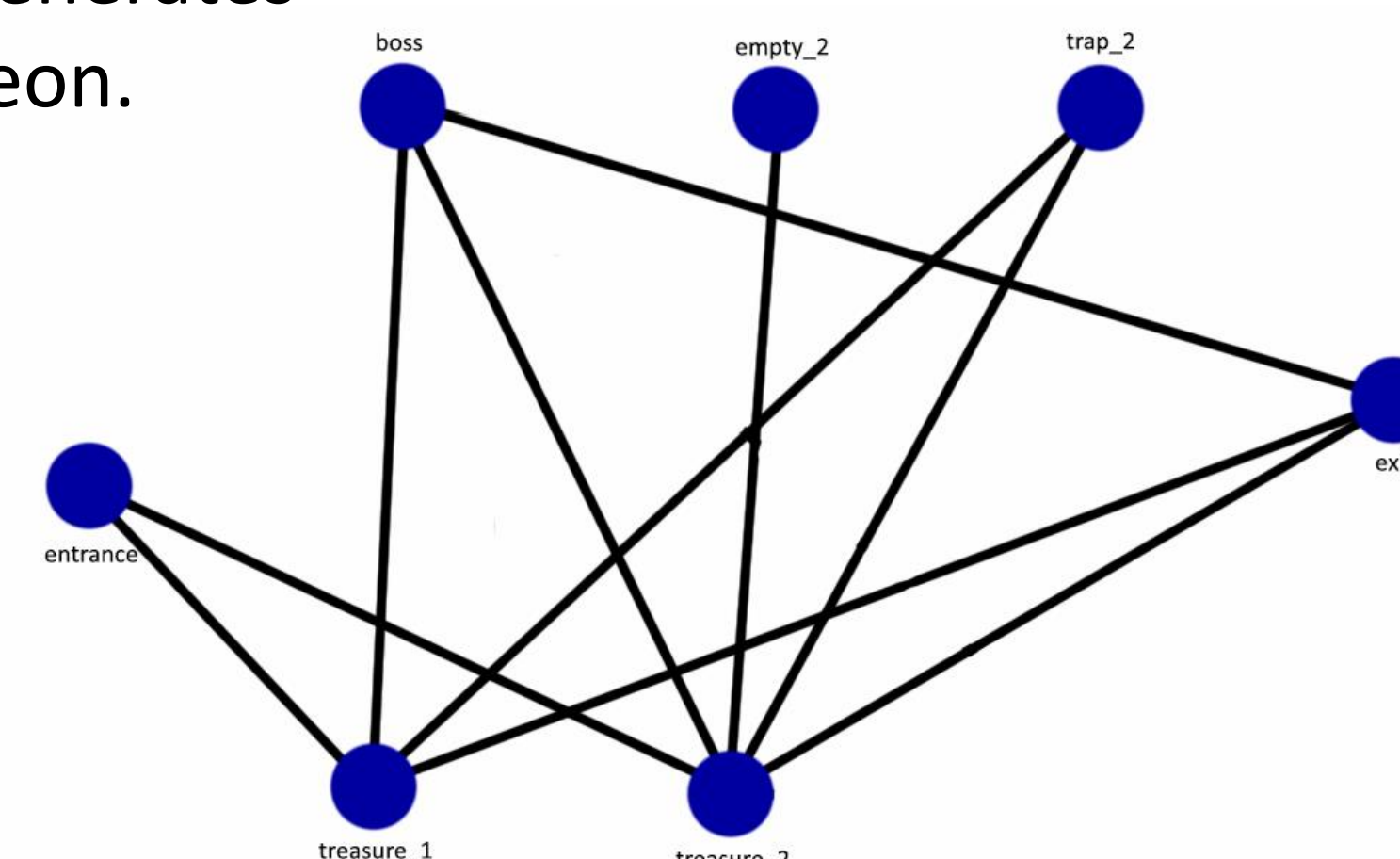
Intervention:

The user can choose parts of the dungeon to regenerate.



Regeneration:

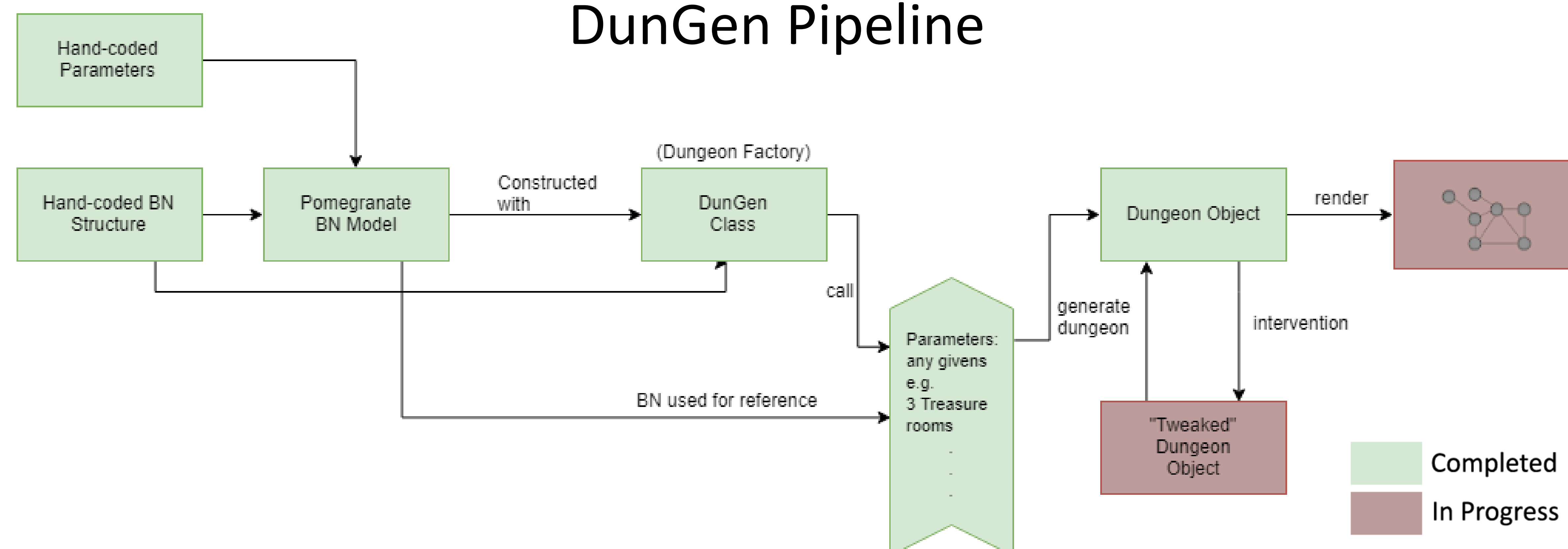
DunGen regenerates a new dungeon.



Areas for Improvement

- The CBN conditional probability tables (CPTs) were worked out by hand and need adjusting to improve the created dungeons' layouts.
- Improve documentation.
- Code optimizations.

DunGen Pipeline



Future Work

DunGen currently only performs traditional Bayesian inference and generates object representations for dungeons. Next steps in the project involve finishing the pipeline to render a 2D dungeon layout in image form and implementing the counterfactual inference cycle.

DunGen was originally envisioned as part of a pipeline to generate fully functional 3D dungeons. Future work could be dedicated to infusing the generated dungeons with room-specific functionality (e.g. keys, levers, switches, etc.), and extending the render pipeline to the generation of 3D dungeon architecture.

References

Merrell, Paul, et al. "Computer-Generated Residential Building Layouts." *ACM Transactions on Graphics*, vol. 29, no. 6, 2010, p. 1., doi:10.1145/1882261.1866203.

