Creative Procedural Content Generation via Machine Learning

Matthew Guzdial
Procedural Content Generation

Matthew Guzdial
Procedural Content Generation
Procedural Content Generation
Generated Content

- Structure
- Story
- Quests
- Art assets
- Music
- Etc...
Levels

Diablo 3

Spelunky

Minecraft

Skyrim, Initial Generation

Bloodborne, Chalice Dungeons
Characters, Story, and Quests

Fallen London

The Shrouded Isle

Skyrim, Radial Quests

Shadow of Mordo
The Promise of PCG

There was a myth in the industry 5-10 years ago that PCG could save costs.
Why is PCG so Tough?

Example approach: Grammar

Imagine instead of creating a sculpture from clay you came up with instructions to make it from Lego Blocks.
Drawbacks of traditional PCG

• Requires a design expert
• Requires a CS expert
• Encoding design knowledge time
• Design iterations more costly
What if we could automate encoding the expert design knowledge?
What if we could automate encoding the expert design knowledge?
Procedural Content Generation via Machine Learning
Snodgrass and Ontañón. Learning to generate video game maps using Markov models.
Genetic Algorithms

Dahlskog and Togelius. Patterns as objectives for level generation.
NeuroEvolution

Hoover et al. Composing Video Game Levels with Music Metaphors through Functional Scaffolding
LSTMs

Summerville and Mateas. “Super Mario as a String: Platformer Level Generation via LSTMs”.
Existing Data

• Many techniques for spatial structural content
• Reliant on human effort to translate content to digital form
• Produces output content similar to input content
Wave Function Collapse
You pushed hard to open the heavy bank door.

You put on a joker mask that covered your entire face.

You drove to the bank in the off-peak hour.

You walked into the bank, trying to look normal. Your pulse quickened.

- Wait in the teller line
- Scan the bank
- Look around for a teller
Building Own Corpus

• Less explored

• Does not rely on existing corpus

• Significant effort and requires the same design expertise as traditional PCG
Alternative Training Data

Learn through some secondary domain by converting data into the desired domain
Alternative Training Data: Video

Benefits
- Existing corpora not always accessible
- Contains player-game element interactions
- Everything to fully recreate a game
Game Learning from Video

Authored Game Engine + Levels

Output Gameplay Video

Learned Level Design

Learned Game Engine
# Generating Mario Levels

<table>
<thead>
<tr>
<th>PCG Research</th>
<th>Source of Design Knowledge</th>
<th>Generative Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mawhhorter &amp; Mateas 2010</td>
<td>Human author</td>
<td>Grammar-based</td>
</tr>
<tr>
<td>Launchpad 2011</td>
<td>Human author</td>
<td>Grammar-based</td>
</tr>
<tr>
<td>Kerssemakers et al 2012</td>
<td>Human author</td>
<td>Evolutionary</td>
</tr>
<tr>
<td>Shaker et al 2012</td>
<td>Human author</td>
<td>Evolutionary</td>
</tr>
<tr>
<td>Dahlskog &amp; Togelius 2014</td>
<td>Level definitions</td>
<td>Evolutionary</td>
</tr>
<tr>
<td>Snodgrass &amp; Ontañón 2014</td>
<td>Level definitions</td>
<td>Markov chains</td>
</tr>
<tr>
<td>Summerville &amp; Mateas 2016</td>
<td>Level definitions</td>
<td>LSTM</td>
</tr>
<tr>
<td>Summerville &amp; Mateas 2016</td>
<td>Gameplay Video</td>
<td>Clustering/Probabilistic Models</td>
</tr>
</tbody>
</table>


System Overview

Deriving Level Chunks

Categorizing Level Chunks

Probabilistic Model
Deriving Level Chunks
G : Geometric information for shapes of sprite $t$
D: Relationships between sprite shape $t$ and all other sprite shapes
N: Counts for sprites found in initial data set.

- block: 22
- ground: 17
- question-block: 5
- goomba: 2
S: Styles of sprite shapes of type $t$
L: Level chunk styles (the styles of sprite shapes and the templates they can create).
Probability Distribution

Build the probability distribution $P(g_{s1}, r_d | g_{s2})$ for each L node.

Using Baye’s law get $P(g_{s2} | g_{s1}, r_d)$ and generate.
Generation
Level Generation
Human Subjects Study

• Online study
• Players played 3 of 16 levels
  – 1 Mario, 5 Snodgrass, 5 Dahlskog, 5 Our System
• Asked to rank the levels on the following traits
  – Mario-like (Style)
  – Fun
  – Frustration
  – Challenge
  – Design
  – Creativity
Question 3: Rank Frustration

For all three levels please rank them from 1-3 (using 1, 2 and 3 in the boxes below) from most to least frustrating with the most frustrating being ranked as "1".

Mario Level Rank       Test Level 1 Rank       Test Level 2 Rank

Reasoning (optional)

Next Question
## Study Results

<table>
<thead>
<tr>
<th></th>
<th>Mario</th>
<th>Ours</th>
<th>Snodgrass</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mario-like</td>
<td>N/A</td>
<td>1</td>
<td>2</td>
<td>0.0174</td>
</tr>
<tr>
<td>Fun</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3.02e-5</td>
</tr>
<tr>
<td>Frustration</td>
<td>3</td>
<td>2</td>
<td>1.5</td>
<td>1.06e-11</td>
</tr>
<tr>
<td>Challenge</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>0.6976</td>
</tr>
<tr>
<td>Design</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2.31e-7</td>
</tr>
<tr>
<td>Deaths</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>2.25e-9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Mario</th>
<th>Ours</th>
<th>Dahlskog</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mario-like</td>
<td>N/A</td>
<td>1</td>
<td>2</td>
<td>0.0383</td>
</tr>
<tr>
<td>Fun</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>0.3147</td>
</tr>
<tr>
<td>Frustration</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>3.58e-7</td>
</tr>
<tr>
<td>Challenge</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2.31e-4</td>
</tr>
<tr>
<td>Design</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2.21e-5</td>
</tr>
<tr>
<td>Deaths</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>2.42e-5</td>
</tr>
</tbody>
</table>
Study Results Summary

- Our system was significantly more similar to Mario than both other generators in the rankings for:
  - Mario-like (style)
  - Design
  - Frustration
  - # of deaths
- Significantly more fun than Snodgrass
- Significantly less challenging than Dahlskog
## Comparison to System Ranking Results

<table>
<thead>
<tr>
<th>Category</th>
<th>$r_s$</th>
<th>$p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Style</td>
<td>0.6115095</td>
<td>2.2e-16</td>
</tr>
<tr>
<td>Design</td>
<td>0.51948</td>
<td>2.2e-16</td>
</tr>
<tr>
<td>Fun</td>
<td>0.2729658</td>
<td>3.745e-5</td>
</tr>
<tr>
<td>Frustration</td>
<td>-0.4393904</td>
<td>6.79e-12</td>
</tr>
<tr>
<td>Challenge</td>
<td>-0.387222</td>
<td>2.351e-09</td>
</tr>
<tr>
<td>Creativity</td>
<td>-0.1559725</td>
<td>0.02007</td>
</tr>
</tbody>
</table>
Conclusions

• We learn a model of level design that correlates strongly with measures of human models of level design.

• These models can produce levels of consistently higher quality than the levels of other machine learned models.
Creative Procedural Content Generation via Machine Learning
Modern Machine Learning

• Good when we have large amounts of data

• Of a form that consistently maps expected input to expected output

• But what about cases where no training data exists? Or where not one optimum output?
  – e.g. most design problems
What we’d like...
Combinational Creativity

Input 1: Resident, Land, Lives in House, Rides on Boat
Input 2: Passenger, Water, Rides on Boat
Concept Blend: Resident/Pasenger, Water, Rides/Lives in House/Boat
Amalgam: Resident, Land, Lives in Boat
Compositional Adaptation: Water, on House
Combinatorial Creativity with ML

1. Show we can take ML models as input
2. Demonstrate recombination of ML models
3. Evaluate recombined model
Has someone done this before?

(Nope)
Pitch

Try this with the game level models we discussed learning and evaluating before.
S-Structure Graph

1. Select L Node
1. Select L Node

2. Extract S Nodes
3. Select all Relationships
3. Select all Relationships
4. Build S-Structure Graph
4. Build S-Structure Graph
4. Build S-Structure Graph
Blending Graphs
Case Study: Lost Levels

• From study we know that human players don’t seem to rank creativity consistently
• But we want to evaluate how well these blended models perform.
• Instead: how well the blended models evaluate “real” blends
Four Types of Models

SMB Model
- Original, non-blended L-nodes

Level-Type Model
- L-nodes that match human-annotated “types” (e.g. “Underwater”, “Castle”)

Blended Model
- Blended L-nodes that best understand target level

Full Blended Model
- All blended combinations of each level type model
Four Types of Models

**SMB Model**
- Original, non-blended L-nodes

**Level-Type Model**
- L-nodes that match human-annotated “types” (e.g. “Underwater”, “Castle”)

**Blended Model**
- Blended L-nodes that best understand target level

**Full Blended Model**
- All blended combinations of each level type model
Conclusions

• Concept blending of (at least some) machine learned models can produce new ML models without additional training

• Concept blending appears to be a reasonable approx. of some human creative processes.
Beyond Blending?

Amalgams  Blends  Compositions  Random
Takeaways

• PCGML is a promising research direction toward the automated generation of content

• Current research directions:
  – Training data
  – Creative output
  – How to make it work for designers
Thanks!

mguzdial3@gatech.com