Constructive Procedural Content Generation

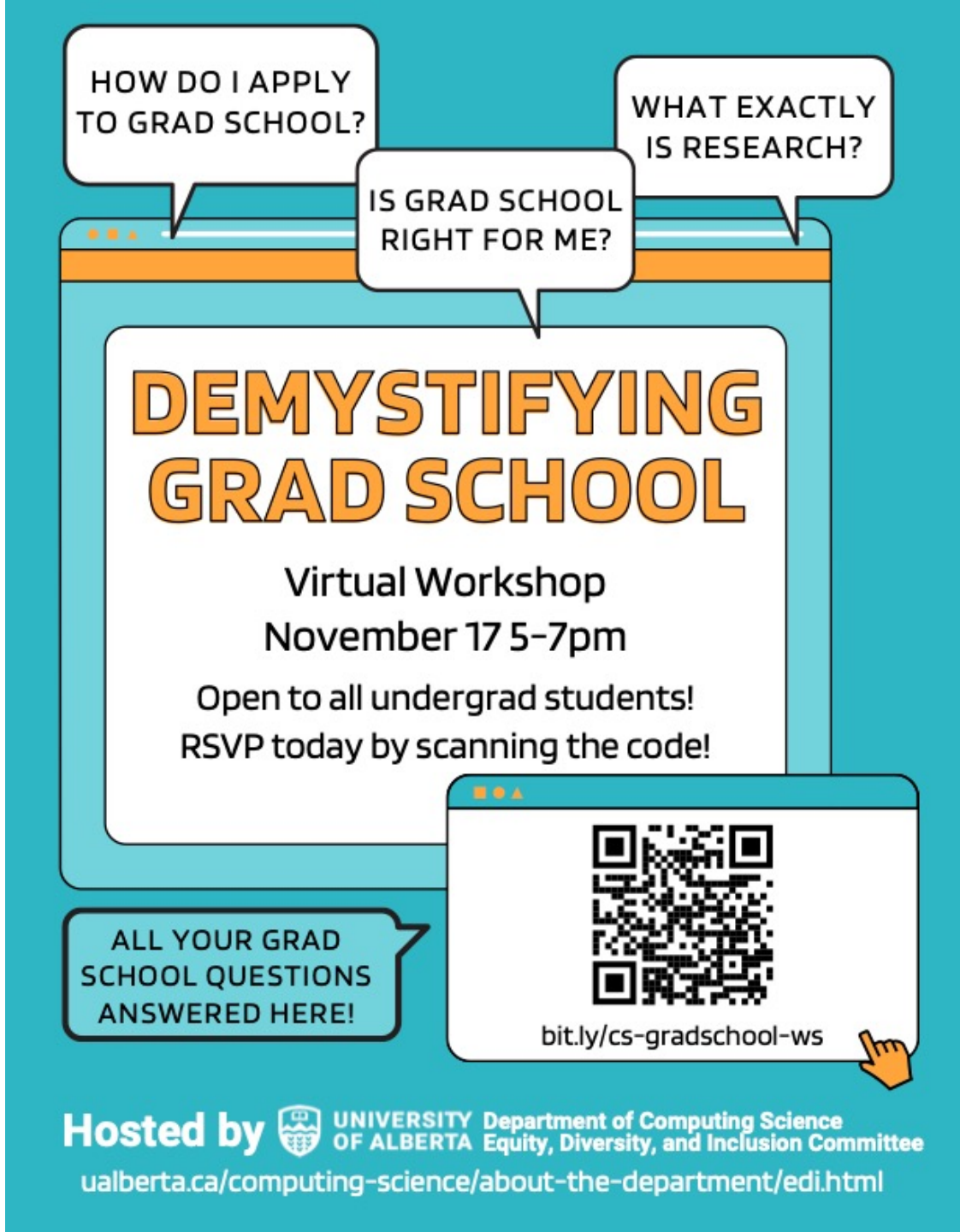
Matthew Guzdial

guzdial@ualberta.ca

Announcements

- Assignment 4 due (make-up exams possible before Dec 7)
- Today: Quiz 4 Review
- Future of Game AI Topic Voting! (next)
- Friday: Practice Quiz
- Assignment 4 passed through autograder by early next week.

Demystifying Grad School
Today! 5-7pm
bit.ly/cs-gradschool-ws



Future of Game AI Topic Voting

https://forms.gle/F1Uxscd5MZq4xig39

https://tinyurl.com/guz-pq28a

1. Reinforcement Learning in Games (53.9%)
2. Balancing Game AI (42.1%)
3. AI for Game Design (39.5%)
4. Automated Playtesting (36.8%)
5. Automated Game Playing (30.3%)
6. AI-based Game Design (28.9%)
6. PCG via Machine Learning (28.9%)
8. Generated Dialogue and story (27.6%)

Quiz 4 Review

Review: Search-based PCG (Examples at the end, time permitting)

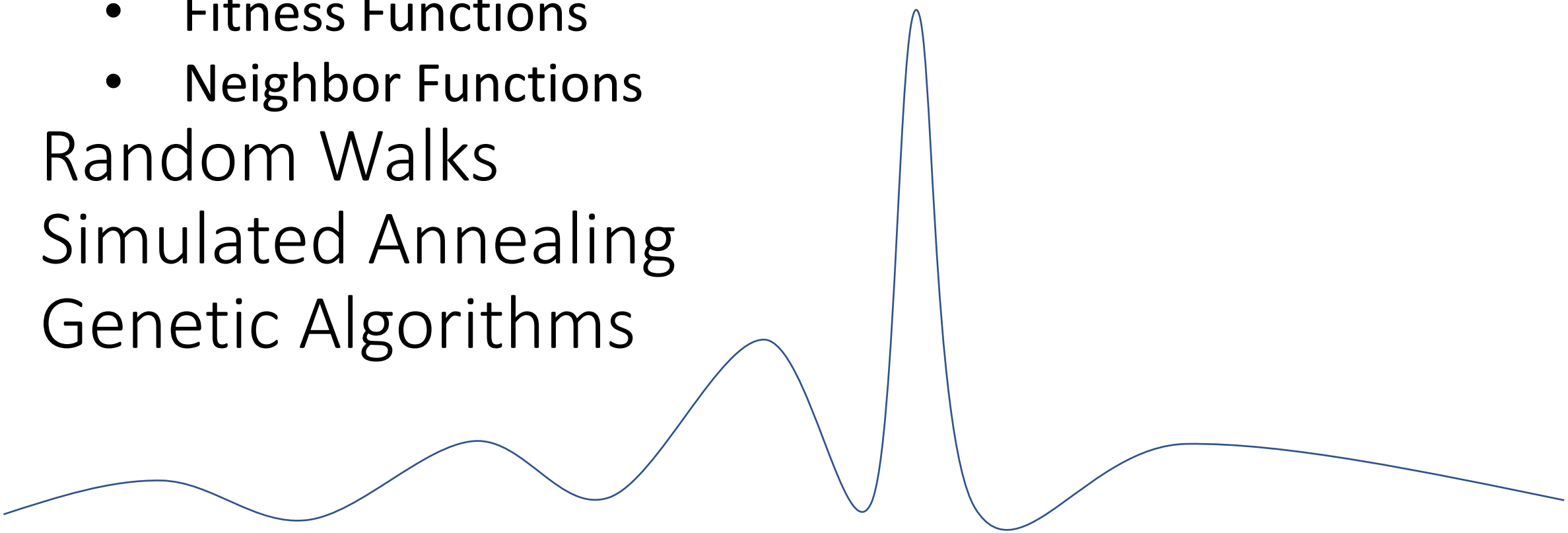
Greedy Hill Climbing

- Fitness Functions
- Neighbor Functions

Random Walks

Simulated Annealing

Genetic Algorithms



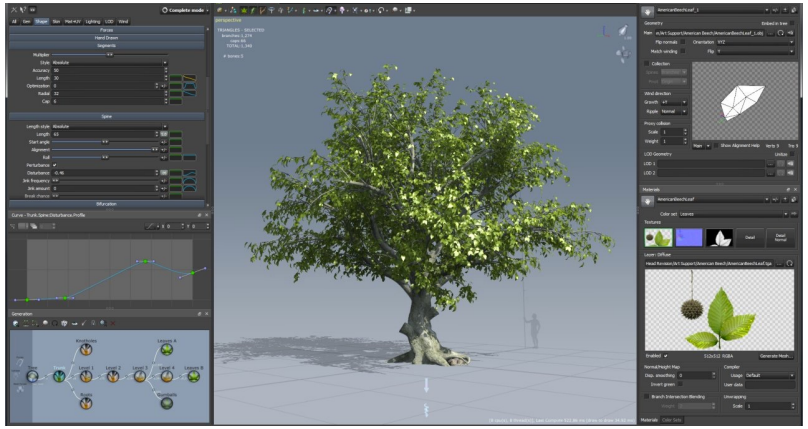
Today: Constructive PCG

- Methods for building up PCG content “piece-by-piece”.
- Slots filled by tokens and constrained by rules.
- By far the most popular PCG method among actual game devs.

Why So Popular?

Compared to Search-based PCG:

1. More controllable
2. Faster
3. More intuitive



Constructive PCG Used For…

Constructive PCG for Game Spaces

Either **Offline** (during development) or **Online** (during the game) generation.

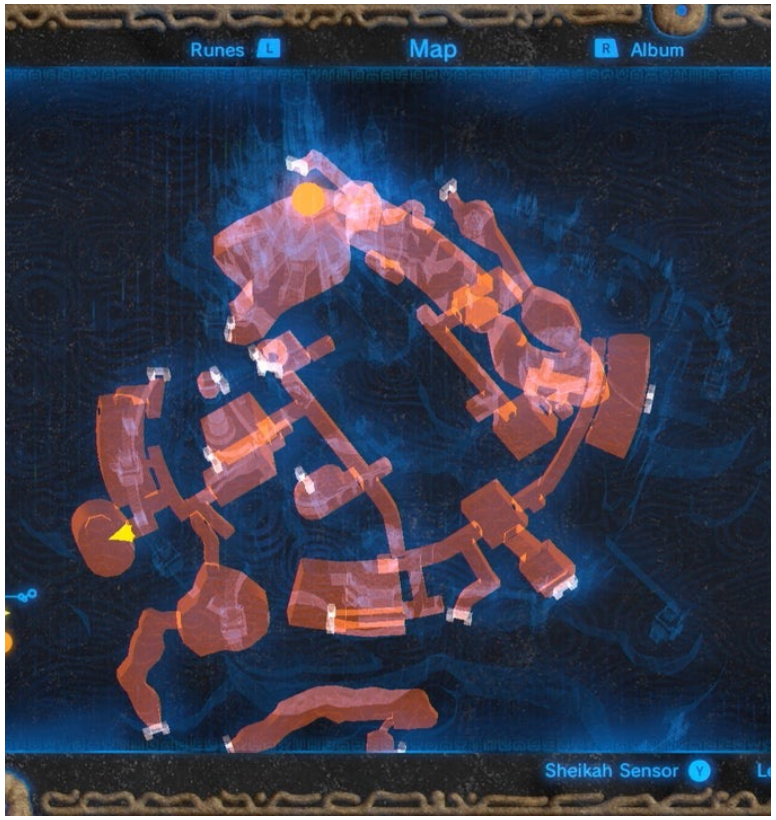
Two major methods:

- **Constructive grammars**: Authoring chunks of a level and rules for how those chunks can be combined.
- **Noise**: General rules for transforming random values into more structured output.

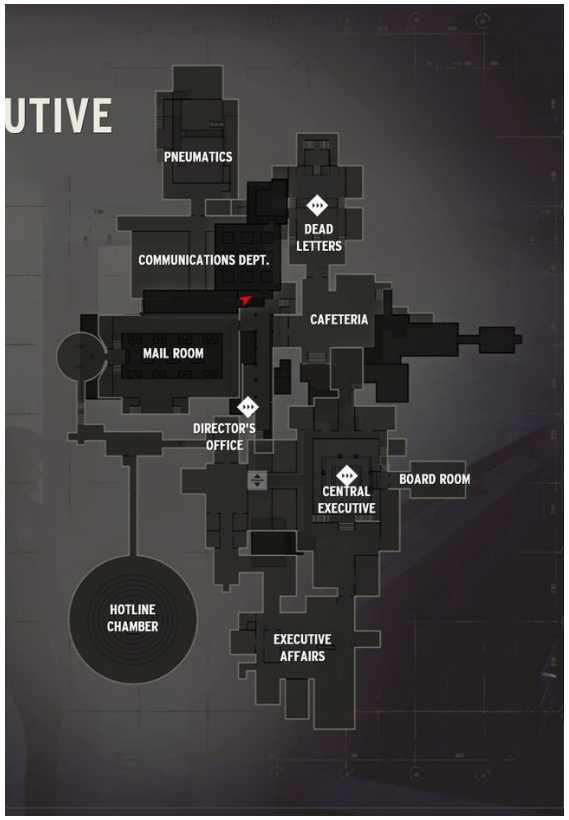
Constructive Grammars

- A type of generative grammar.
- Has a **template** with a series of slots.
- A set of **tokens** that can go into each kind of slot.
- A set of **rules** for how certain **categories** of token interact.

Constructive Grammar:
3D Adventure Game Example



Zelda: Breadth of the Wild

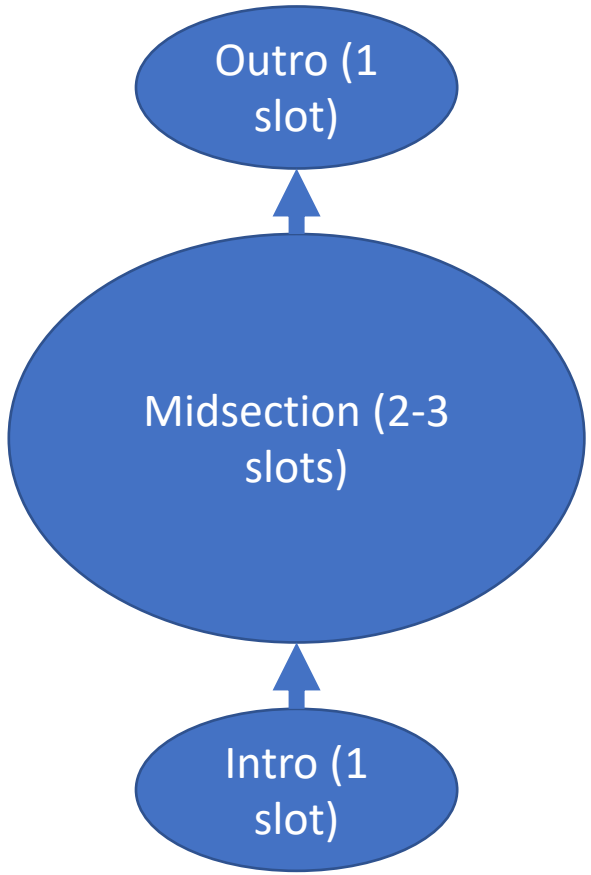


Control

Sekiro: Shadows Die Twice

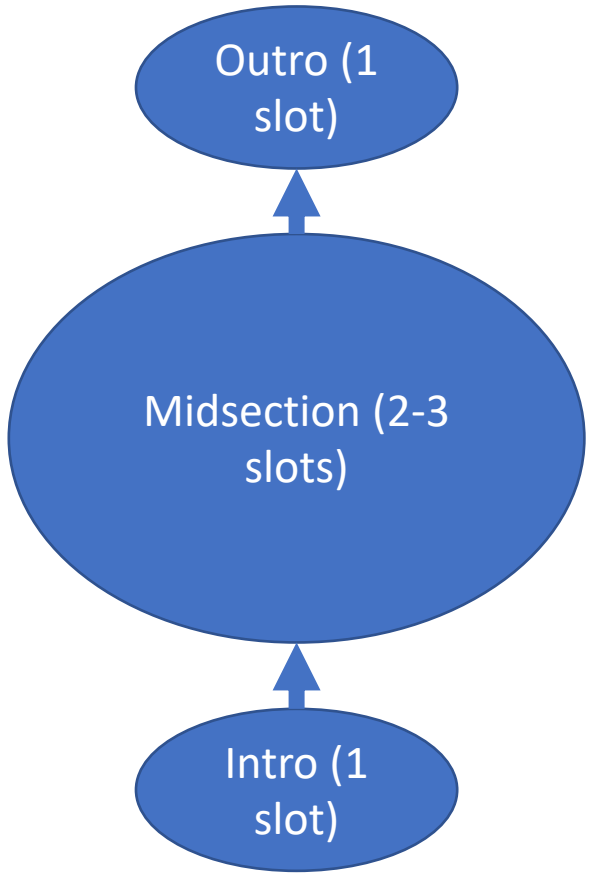
Constructive Grammar:
3D Adventure Game Example

Template

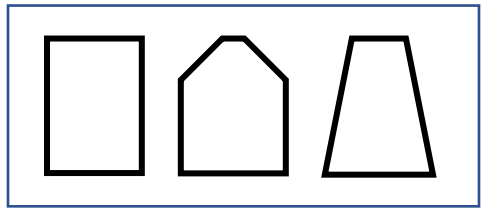


Constructive Grammar:
3D Adventure Game Example

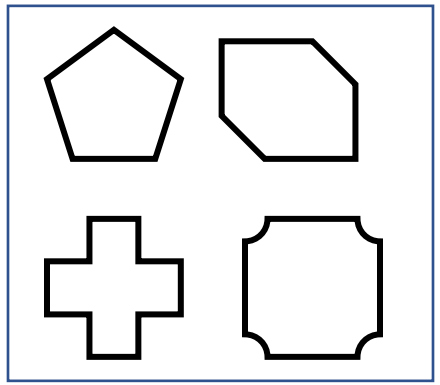
Template



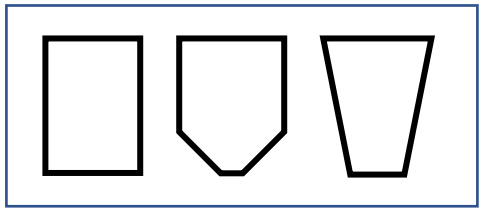
Intro



Midsection

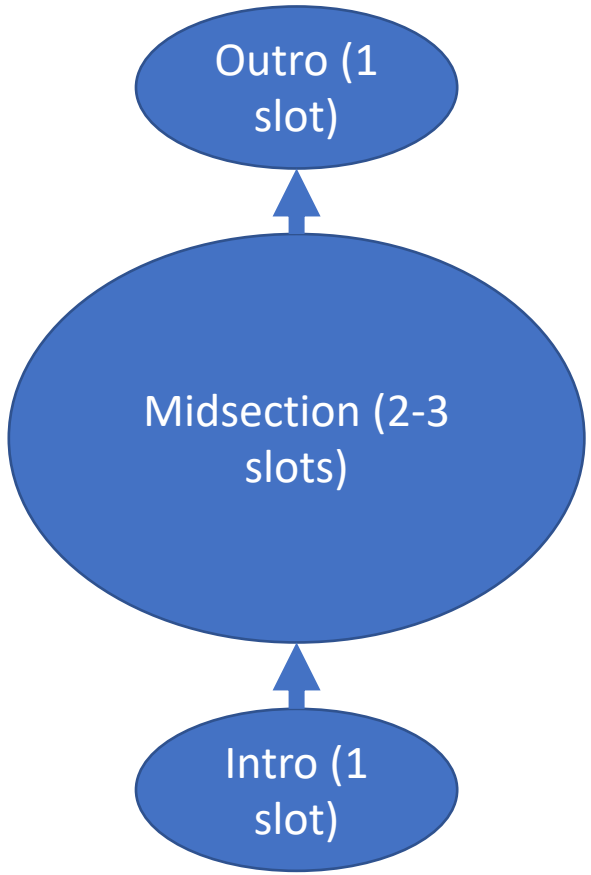


Outro

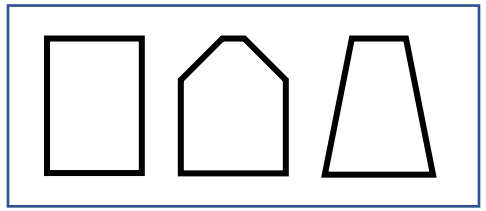


Constructive Grammar:
3D Adventure Game Example

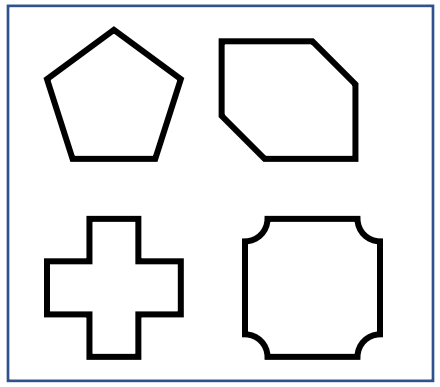
Template



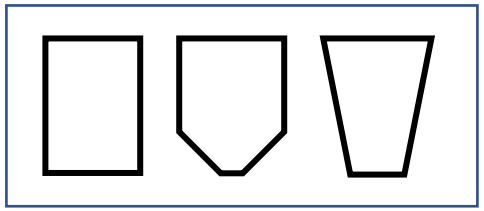
Intro



Midsection



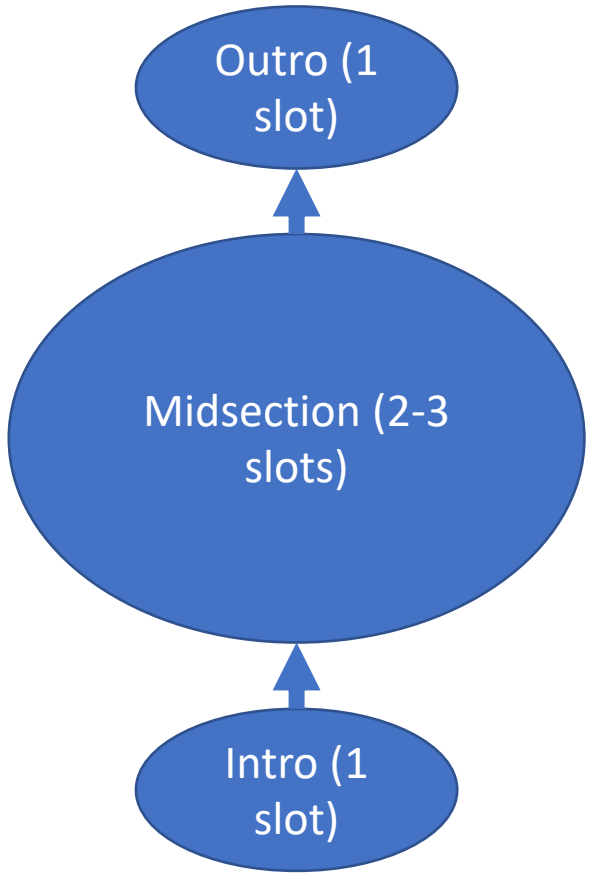
Outro



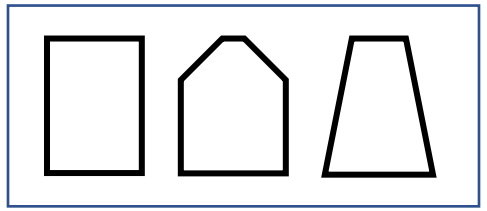
Rules:
-Intro and Outro Connections (any flat surface) must connect to midsections.

Constructive Grammar:
3D Adventure Game Example

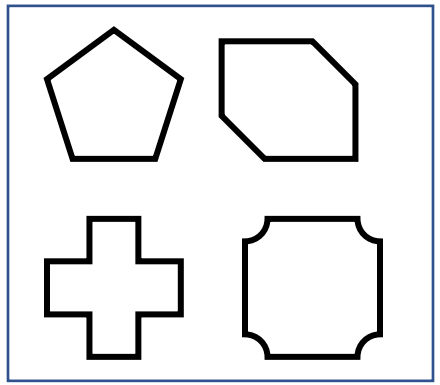
Template



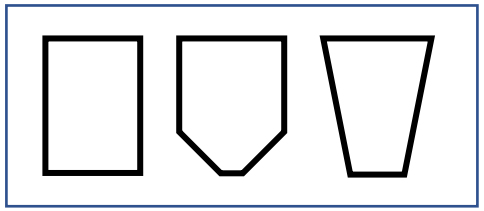
Intro



Midsection

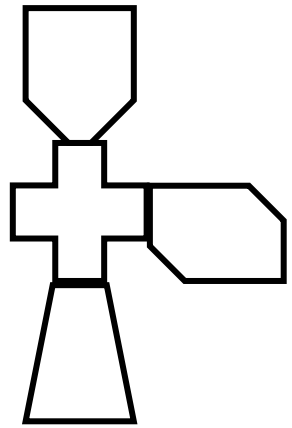


Outro



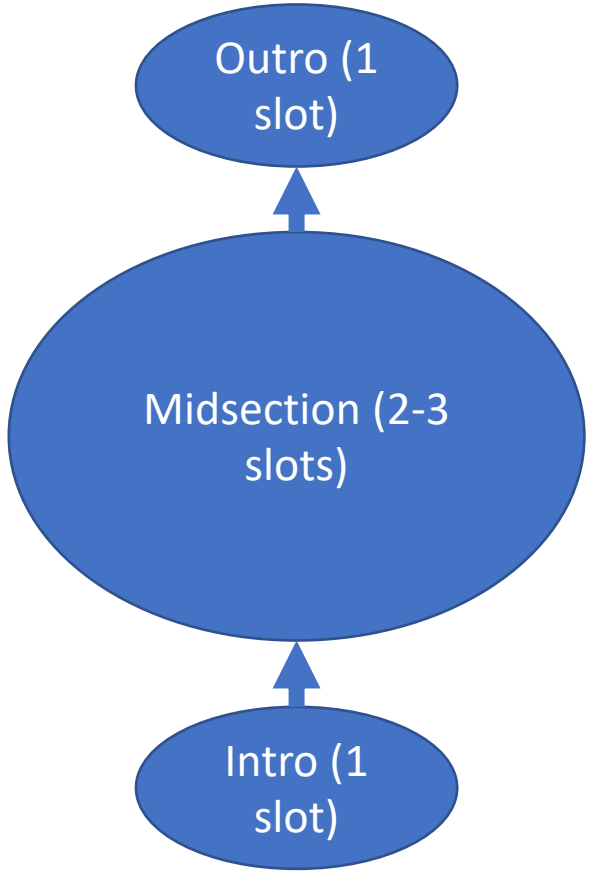
Rules:
-Intro and Outro Connections (any flat surface) must connect to midsections.

<u>Example Output</u>

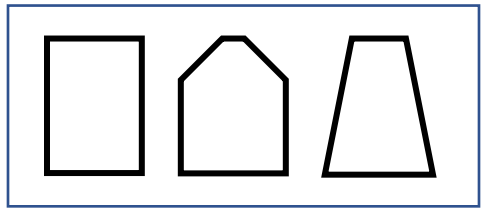


Constructive Grammar:
3D Adventure Game Example

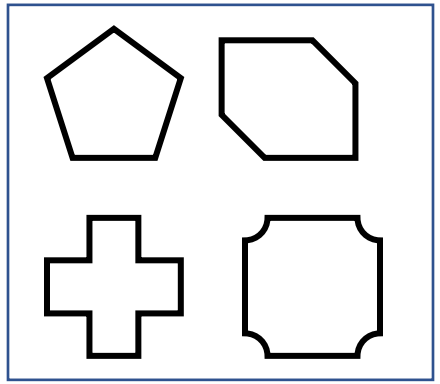
Template



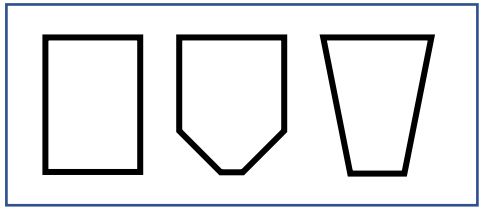
Intro



Midsection



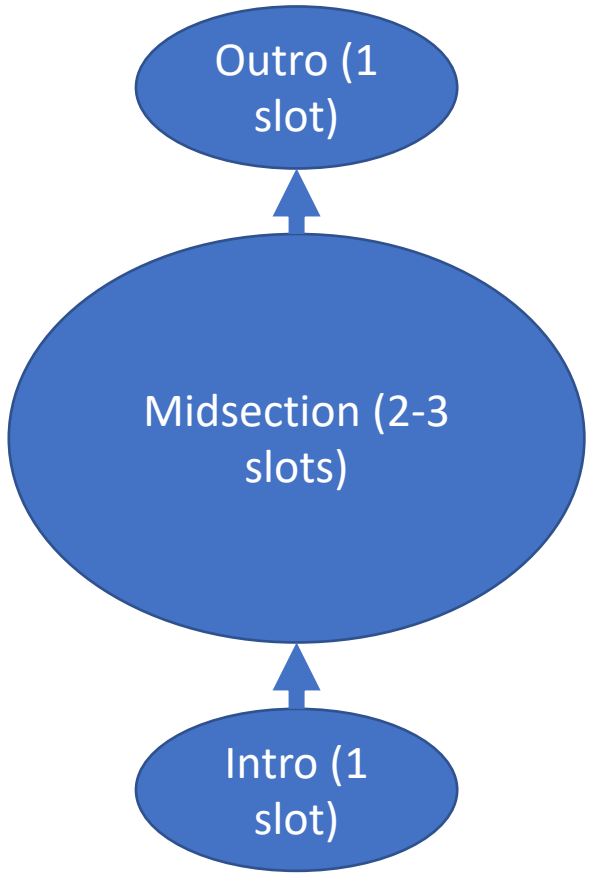
Outro



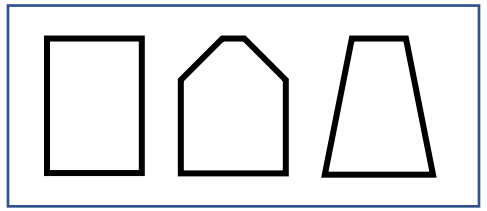
Rules:
-Intro and Outro Connections (any flat surface) must connect to midsections.
-Midsections must be placed vertically.

Constructive Grammar:
3D Adventure Game Example

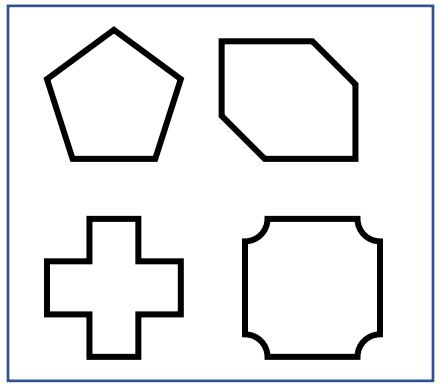
Template



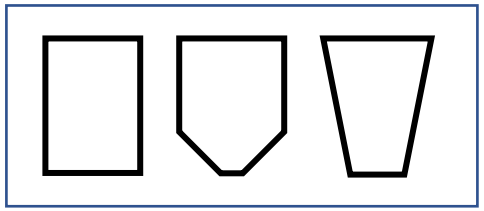
Intro



Midsection

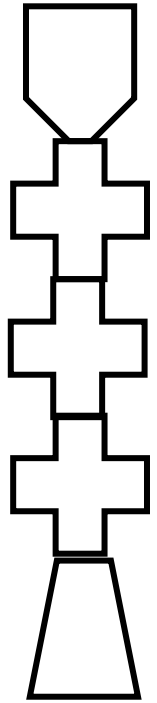


Outro



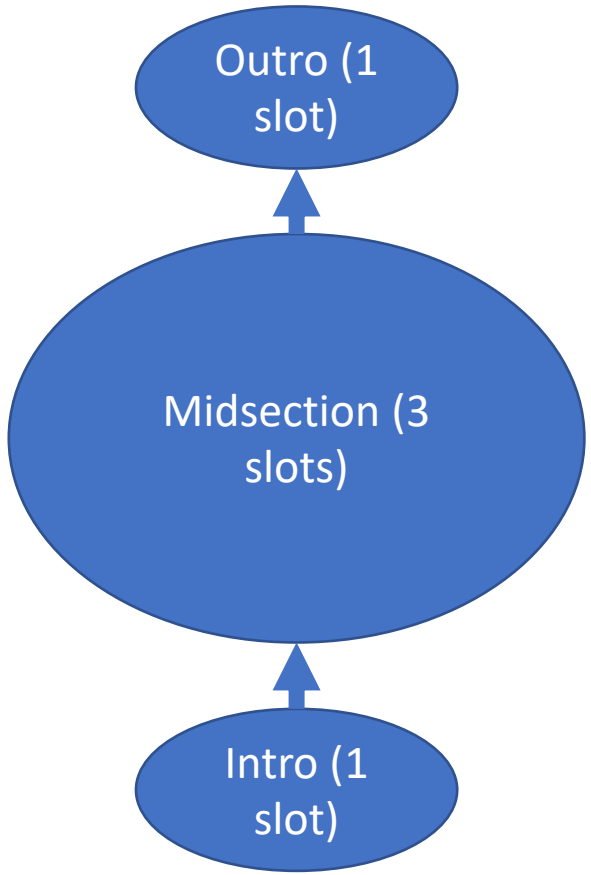
Rules:
-Intro and Outro Connections (any flat surface) must connect to midsections.
-Midsections must be placed vertically.

<u>Example Output</u>

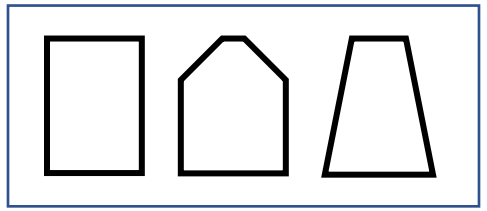


Constructive Grammar:
3D Adventure Game Example

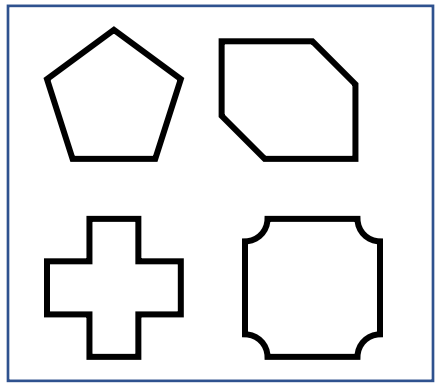
Template



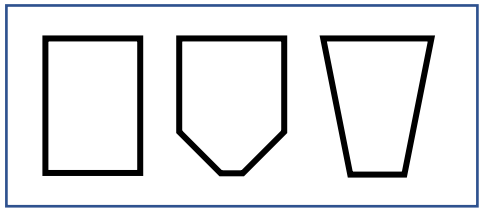
Intro



Midsection



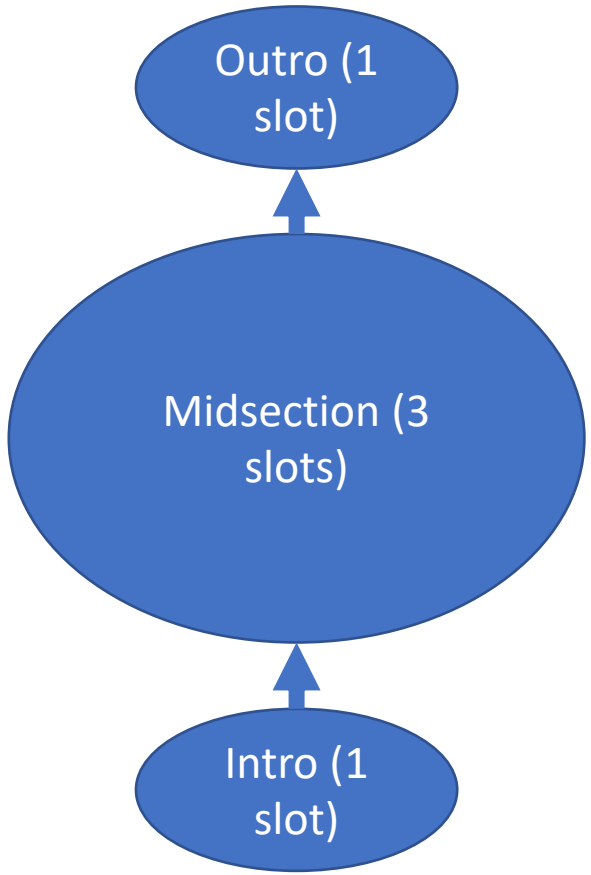
Outro



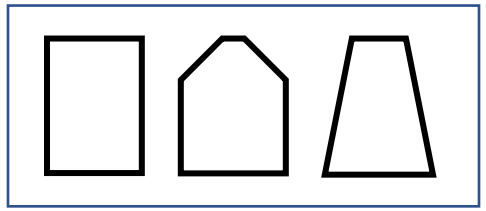
Rules:
-Intro and Outro Connections (any flat surface) must connect to midsections.
-Midsections must have at least one vertical section and one horizontal section.

Constructive Grammar:
3D Adventure Game Example

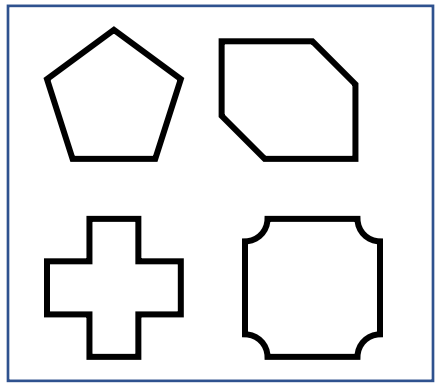
Template



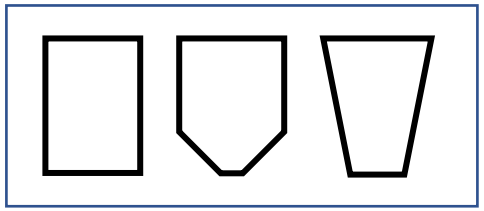
Intro



Midsection

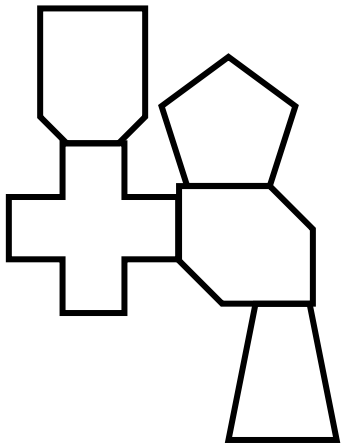


Outro



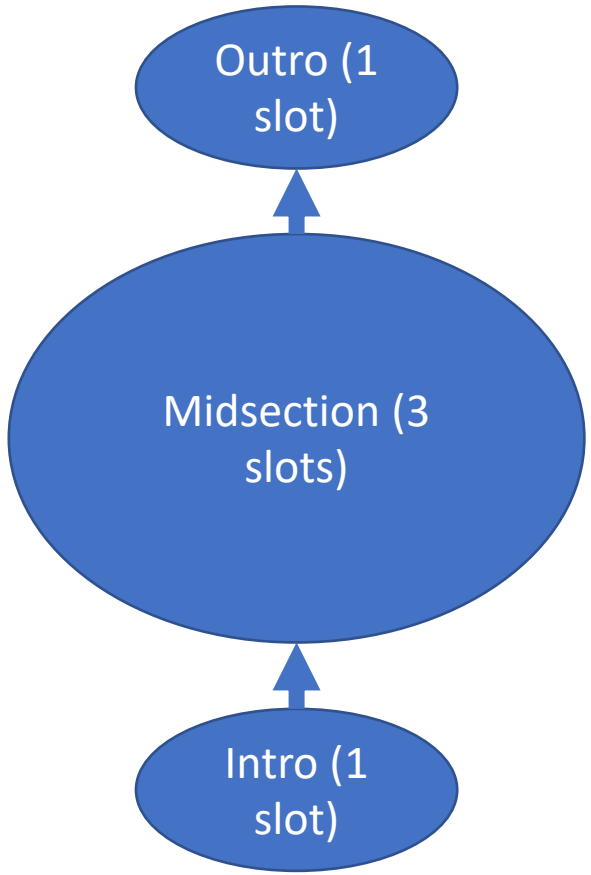
Rules:
-Intro and Outro Connections (any flat surface) must connect to midsections.
-Midsections must have at least one vertical section and one horizontal section.

<u>Example Output</u>

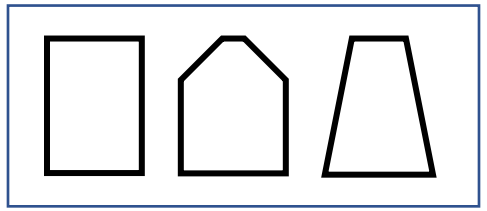


Constructive Grammar:
3D Adventure Game Example

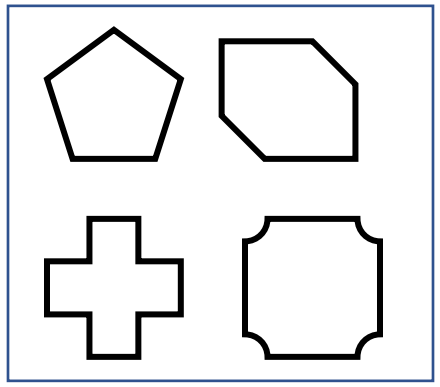
Template



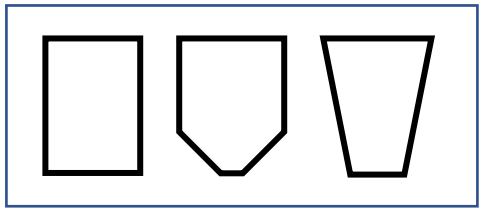
Intro



Midsection



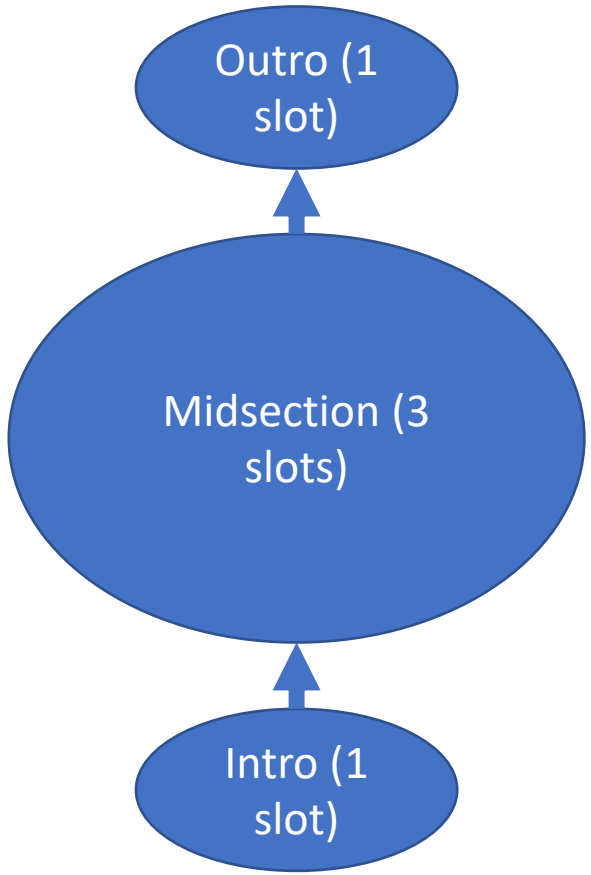
Outro



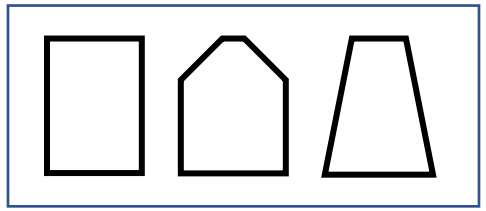
Rules:
-Intro and Outro Connections (any flat surface) must connect to midsections.
-Midsections must have at least one vertical section and one horizontal section.

Constructive Grammar:
3D Adventure Game Example

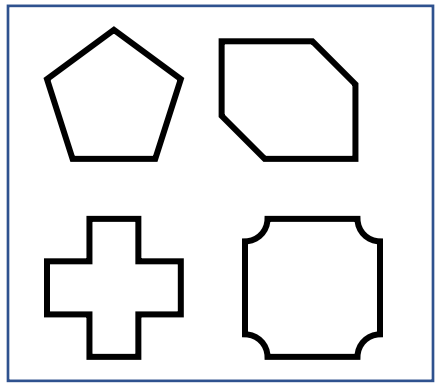
Template



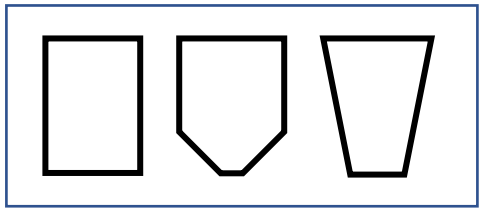
Intro



Midsection

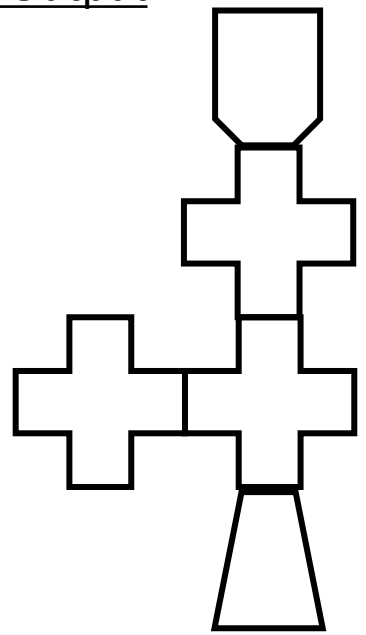


Outro



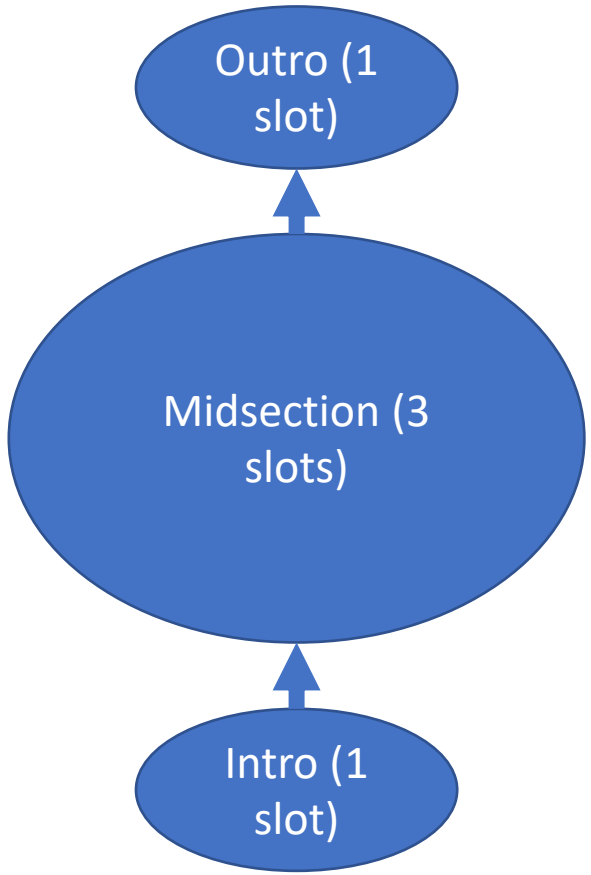
Rules:
-Intro and Outro Connections (any flat surface) must connect to midsections.
-Midsections must have at least one vertical section and one horizontal section.

Example Output

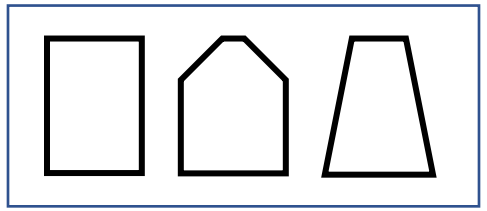


Constructive Grammar:
3D Adventure Game Example

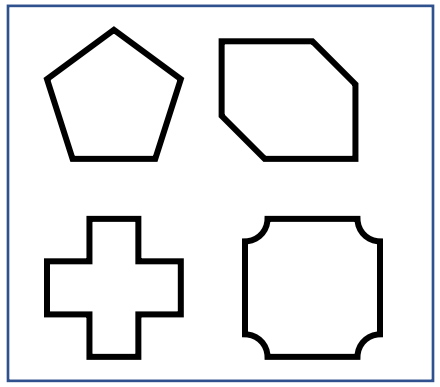
Template



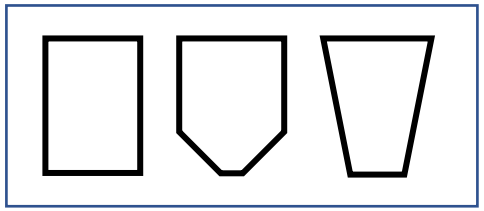
Intro



Midsection



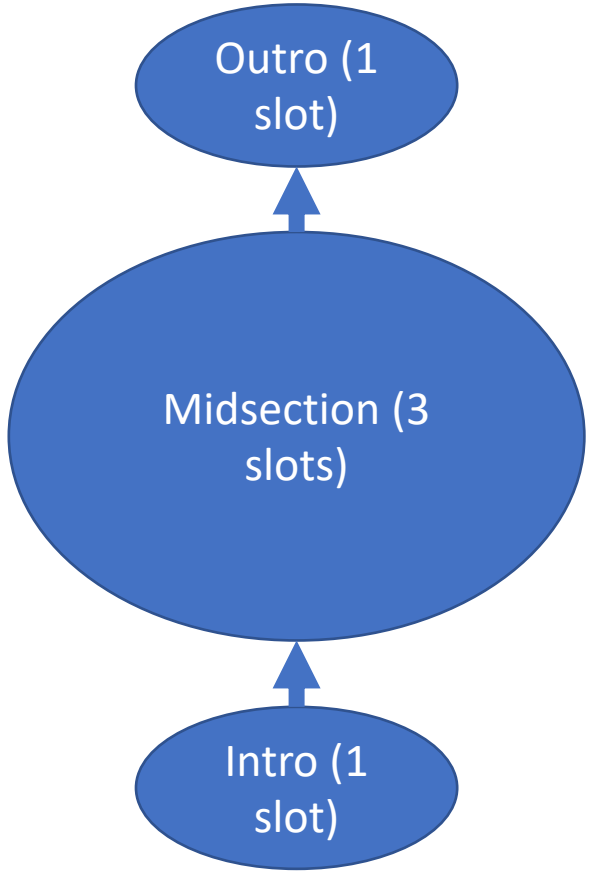
Outro



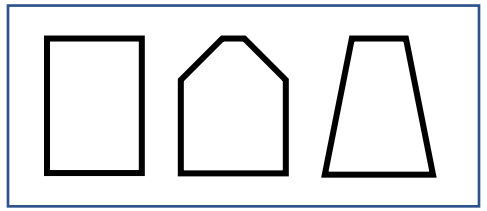
Rules:
-Intro and Outro Connections (any flat surface) must connect to midsections.
-Midsections must have at least one vertical section and one horizontal section.
-Each midsection may only be used once.

Constructive Grammar:
3D Adventure Game Example

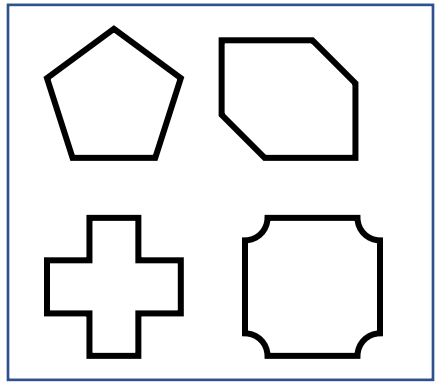
Template



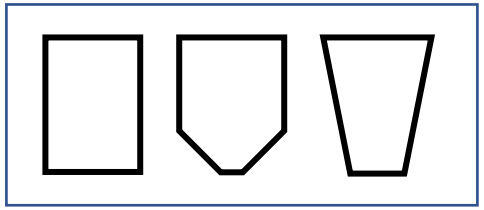
Intro



Midsection

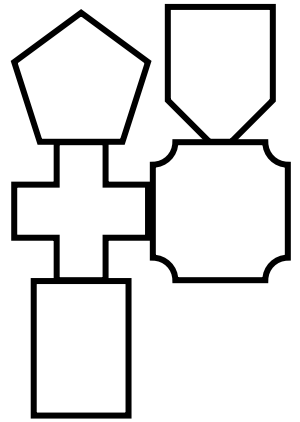


Outro



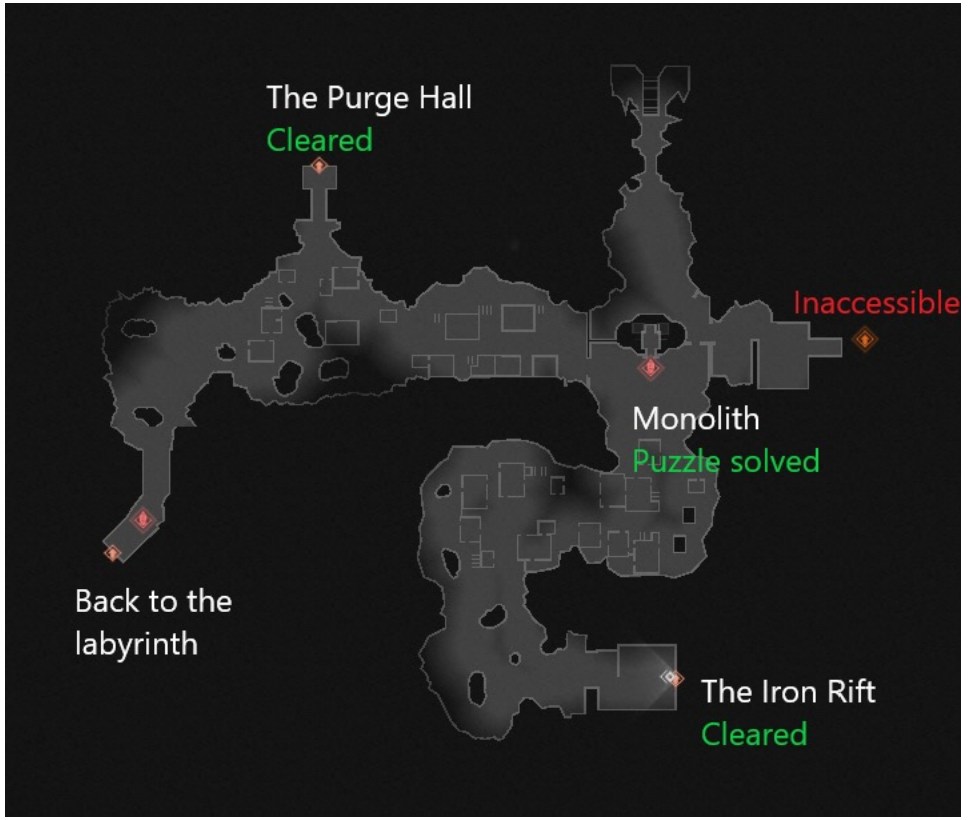
Rules:
-Intro and Outro Connections (any flat surface) must connect to midsections.
-Midsections must have at least one vertical section and one horizontal section.
-Each midsection may only be used once.

<u>Example Output</u>



Downsides of Constructive Grammars

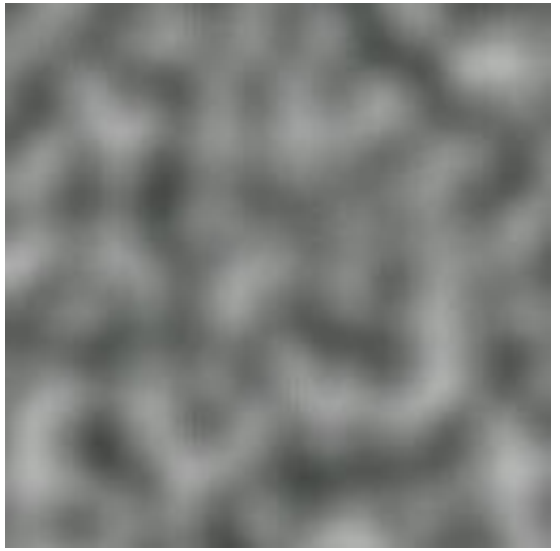
- No guarantee that they will create completable levels.
- Combinatorics makes it functionally impossible to test for all possible output.
- Can take a lot of hand-authoring of tokens to ensure level variety.



Remnant: From the Ashes: PCG

Noise

- Originally developed for texture synthesis.
- Noise takes initially random values on a grid and runs some set of rules over them to give the appearance of structure.
  - Kate Compton “Flavors of Noise”
- Perlin Noise (originally developed for Tron (1982)): Is a favorite for PCG landscape.

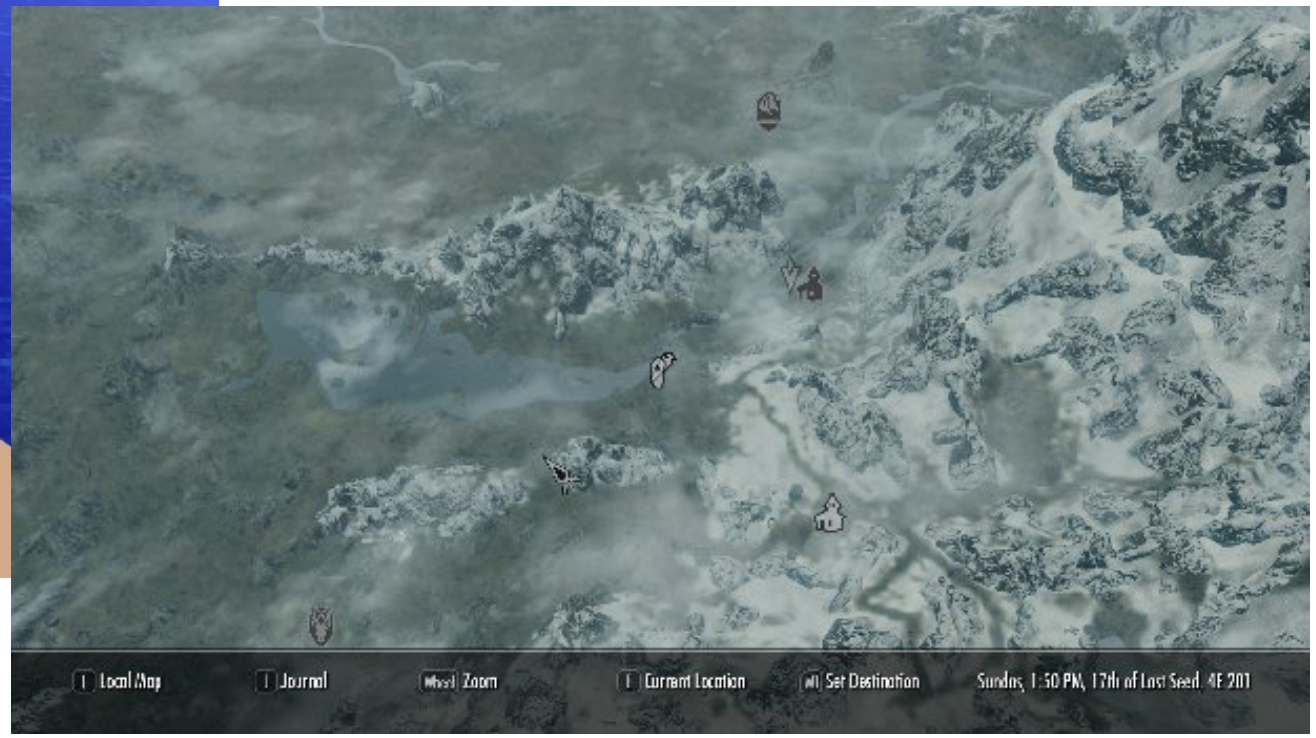


Perlin Noise

Value Noise

Perlin Noise in Games

Minecraft (Online)



Skyrim (Offline)

Noise Unity Demo

Noise

- Can get *something* that can looks okay quite quickly.
- Requires some expertise to know how to adapt to your specific use case.
- Same with grammar, no way to guarantee specific criteria (completability, interestingness, etc.)

Constructive PCG for Game Scenarios

Much the same!

Quest Templates

- Templates: like <LOCATION>, slots that can be filled by specific values
- Tokens: The elements that can fill variables
- Rules: Constraints between variables that limit the types of values we can place in them.

Quest Templates:
Skyrim and Fallout 4

- Collect values of types (LOCATION, NPC, etc) once the player has visited or been made aware of them
- On generation select a template and fill in with values from these types.

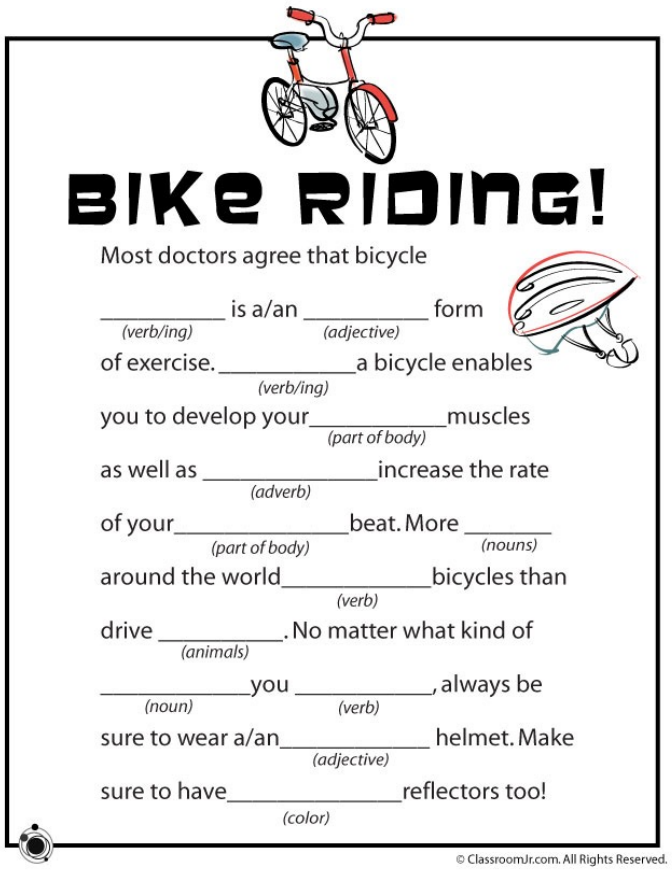
Quest Templates

Pros

- Fast (basically a generative grammar)
- Can be customized to player experience

Cons

- Basically madlibs where you do the same madlib over and over again
- Gets boring quickly



BIKE RIDING!

Most doctors agree that bicycle

_____ is a/an _____ form
(verb/ing) (adjective)

of exercise. _____ a bicycle enables
(verb/ing)

you to develop your _____ muscles
(part of body)

as well as _____ increase the rate
(adverb)

of your _____ beat. More _____
(part of body) (nouns)

around the world _____ bicycles than
(verb)

drive _____. No matter what kind of
(animals)

_____ you _____, always be
(noun) (verb)

sure to wear a/an _____ helmet. Make
(adjective)

sure to have _____ reflectors too!
(color)

Same thing with music!

- Most modern games have a procedural music system, at least for sequencing different pre-authored tracks (very basic grammar).
- More complex approaches use distinct tracks (bass, guitar, percussion etc.) and combine them in different ways according to a simple constructive grammar.

Electroplankton (2005)

https://www.tandfonline.com/doi/pdf/10.1080/0749446080266398 3?casa_token=caiYuhwE0HUAAAAA:bUVKY-i6_y2V9LUXhe7IDtSsHixgrH8NOF5nSTn5bMgB2nrSV2wm54C8Cf6XUhL6JKk9_ik8vGBP

PQ2 (time permitting)
https://tinyurl.com/guz-pq28b
https://forms.gle/k2p2YQsmKQDGnAJL7

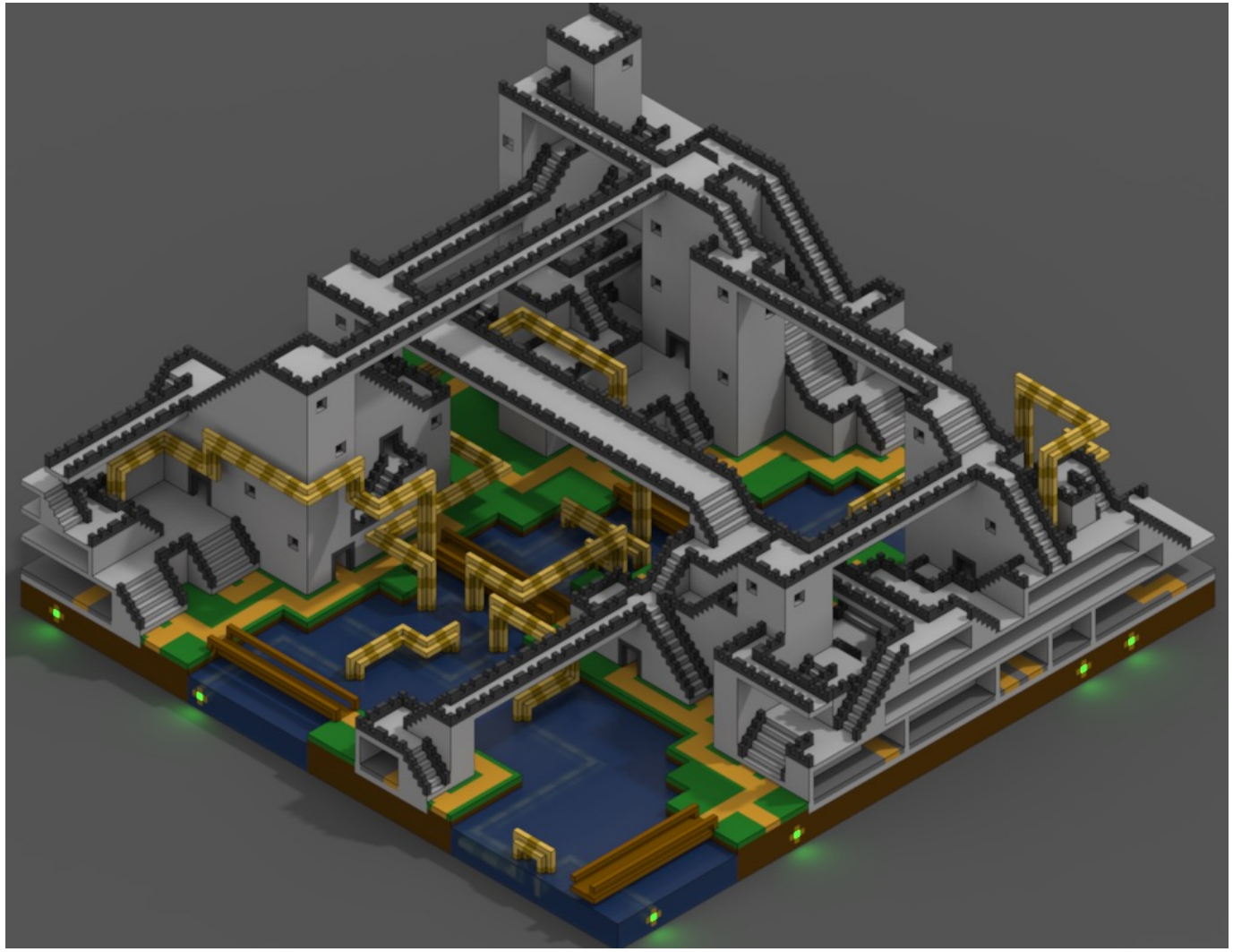
What approach (that we've seen already) could we use instead of constructive grammar/templating for quest or story generation in games? How would this work?

My Answer

- Well really Fred Charles, Marc Cavazza and Steven Mead's answer
- Planning! Specifically Hierarchical Task Networks
- Examples
  - https://www.youtube.com/watch?v=0erFey9hQTY
  - https://www.youtube.com/watch?v=3wzSb8fzDA4

Other example: Elsinore

Friday: Constrained-based PCG
“New” Hotness “Wave Function Collapse”



https://github.com/mxgmn/WaveFunctionCollapse

GA Example (Time Permitting)