

Punch Out Model Synthesis

A Stochastic Algorithm for Constraint Based Tiling Generation

Zzyv Zzyzek

November 19th, 2024

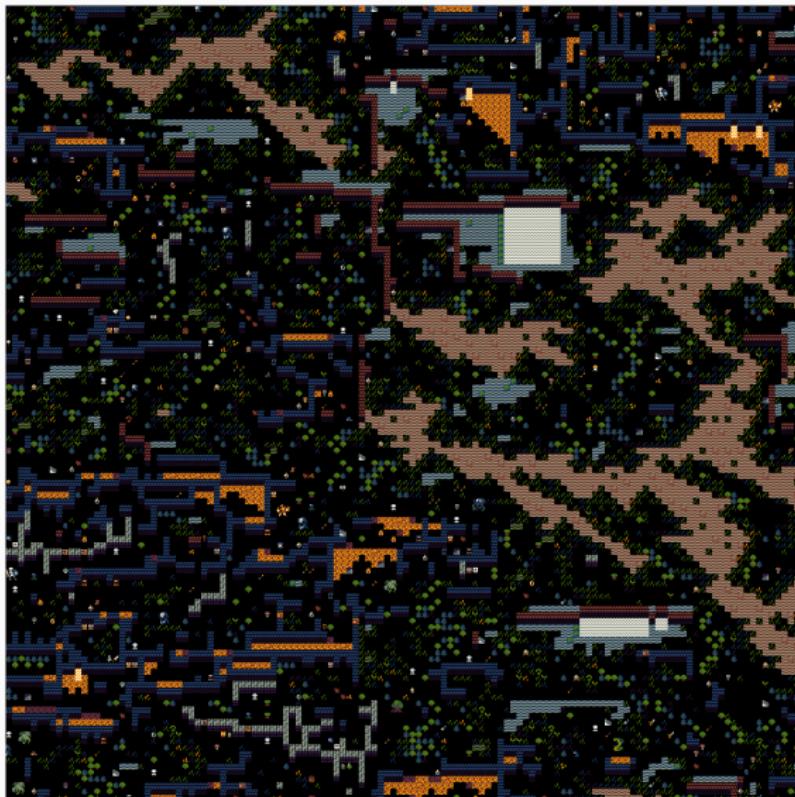
Introduction

Punch Out Model Synthesis (POMS) A Constraint Based Tiling Generation (CBTG) algorithm:

- Works on large grids
- Minimal setup requirements
- Resilience to contradiction

Introduction

Punch Out Model Synthesis (*POMS*)



Introduction

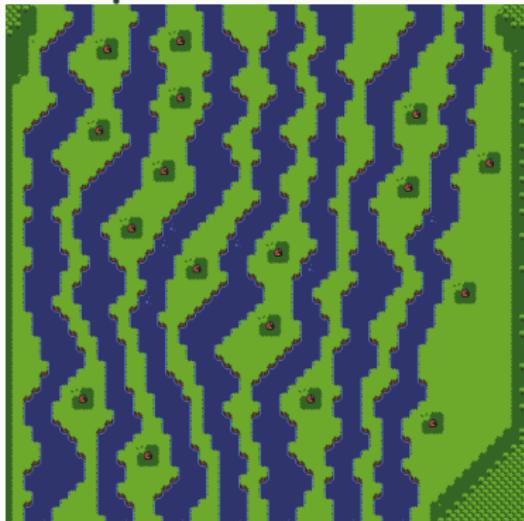
Constraint Based Tiling Generation (*CBTG*) Problem

Find a valid grid realization

A *realization* is a single *tile* placement at each *cell* respecting *constraints*.

(Cell can hold up to D tiles)

Example



Introduction

A region is *Arc Consistent* if all *tiles* in every *cell* within the region have at least one valid neighbor in each direction

The basis for a *Constraint Propagation* algorithm can be made by removing tiles without a valid neighbor

Block Level Solver: completely maintains *Arc Consistency*

Grid Level Solver: only keep minimal information for the entire grid but work on *block* sub-regions

Related Work

	<i>WFC</i>	<i>BMS</i>	<i>MMS</i>	<i>POMS</i>
Solver Type	Block	Block	Grid	Grid
Contradiction	No	Yes	Yes	Yes
Resilience				
Block Step	n/a	n/a	Yes	No
Consistent				
Indeterminate	Yes	Yes	No	Yes
Initial State				
Ergodic	Yes	Yes	No	Yes

WFC: Wave Function Collapse (Gumin)

BMS: Breakout Model Synthesis (Hoetzlein)

MMS: Modify in Blocks Model Synthesis (Merrell)

Related Work

Tile Arc Consistent Correlation Length (TACCL) (Hoetzlein)

How much influence does a tile choice have over long distances?

TACCL as a heuristic to estimate correlation length

Related Work

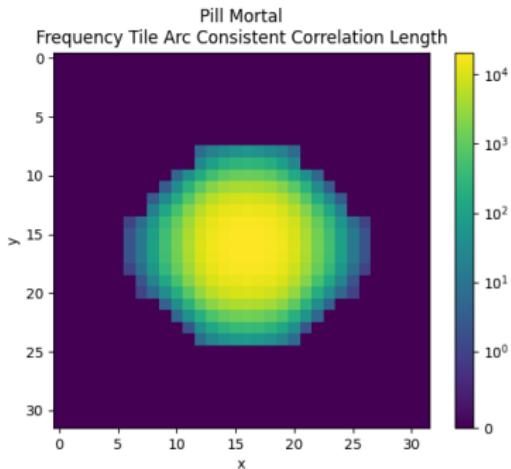
Tile Arc Consistent Correlation Length (TACCL)

TACCL

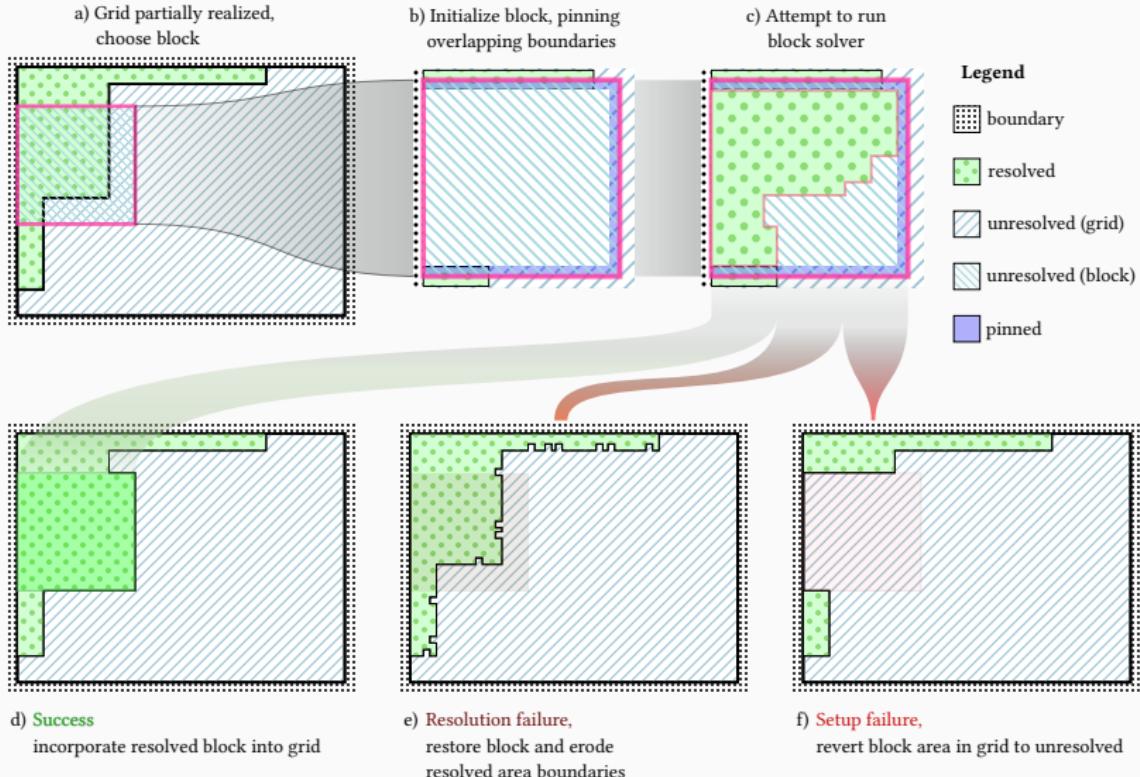
Take block in isolation

- Set block to indeterminate state
- Fix a tile at the center
- Propagate constraints
- Take minimum bounding box of altered cells
- Repeat for all tiles

Example

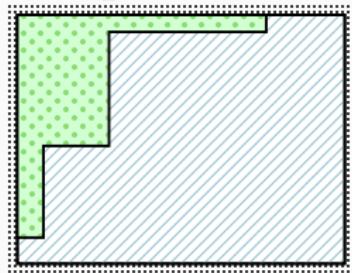


Algorithm: Overview



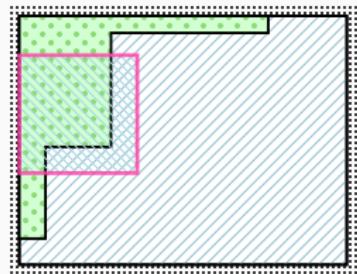
Algorithm

Grid partially realized

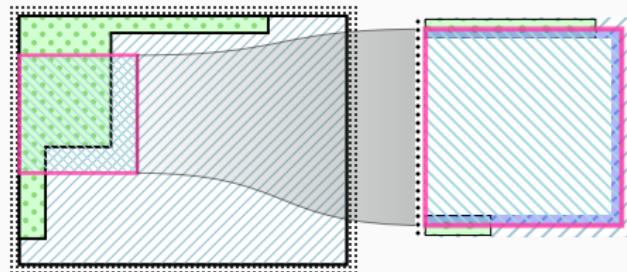


Algorithm

Choose block

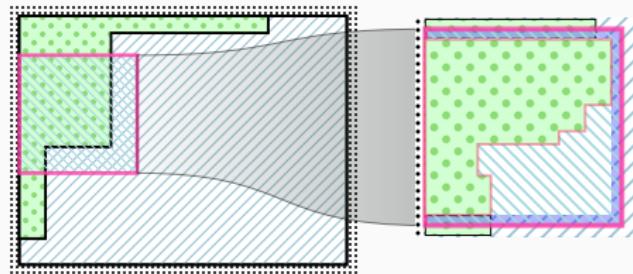


Algorithm

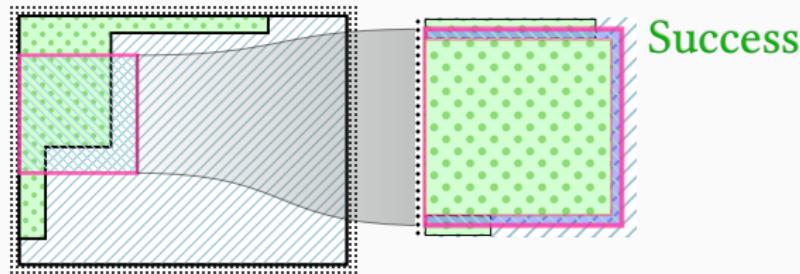


Initialize block
Pin boundaries
Revert interior
(Apply any restrictions)

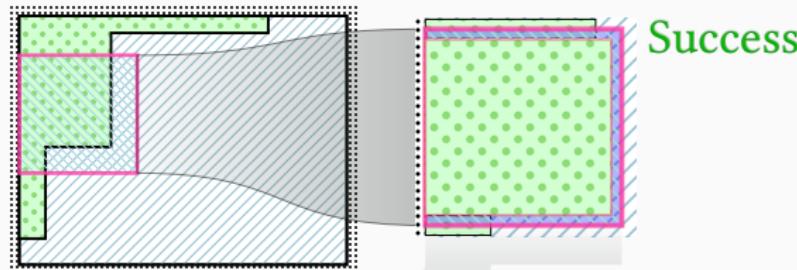
Algorithm



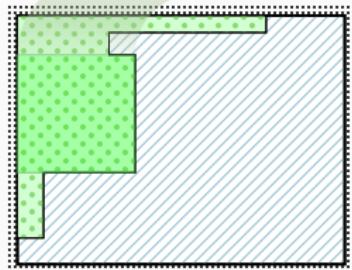
Algorithm



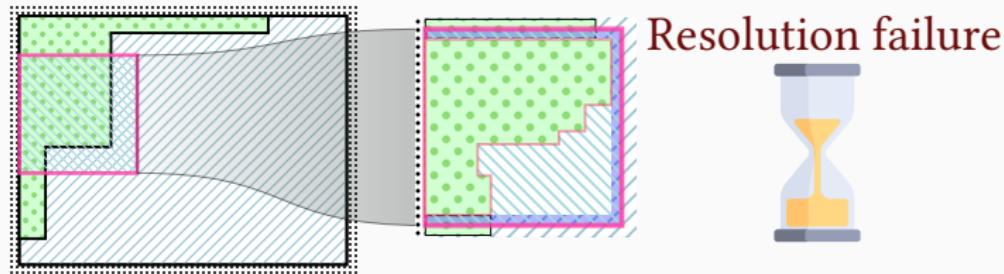
Algorithm



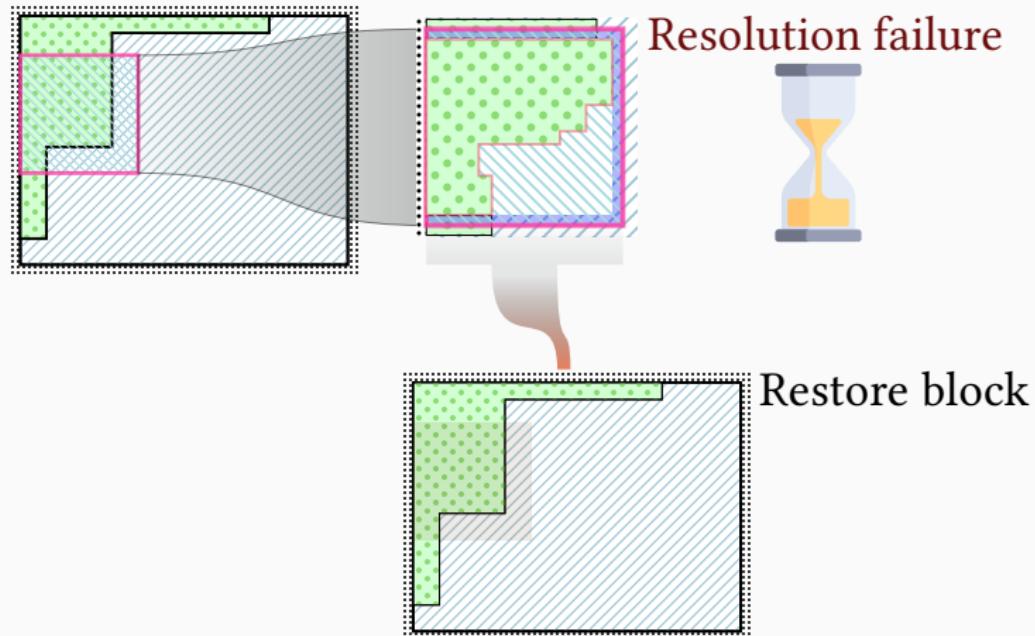
Incorporate resolved
block into grid



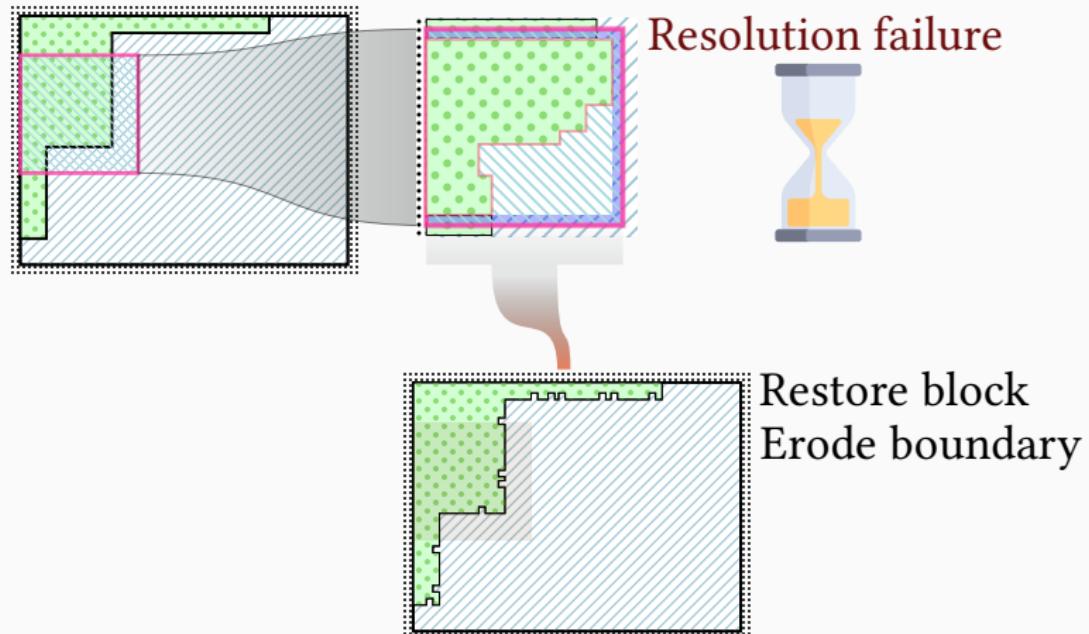
Algorithm



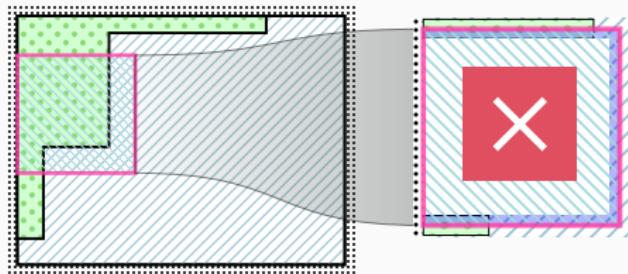
Algorithm



Algorithm



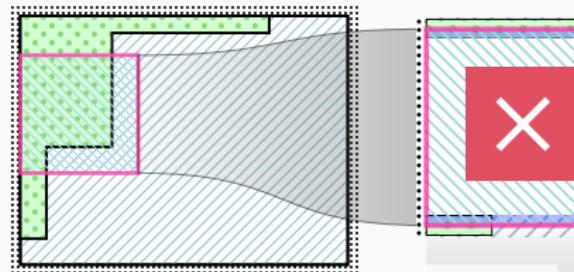
Algorithm



Setup failure

Failed initial
Arc Consistency

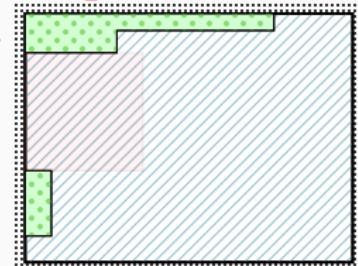
Algorithm



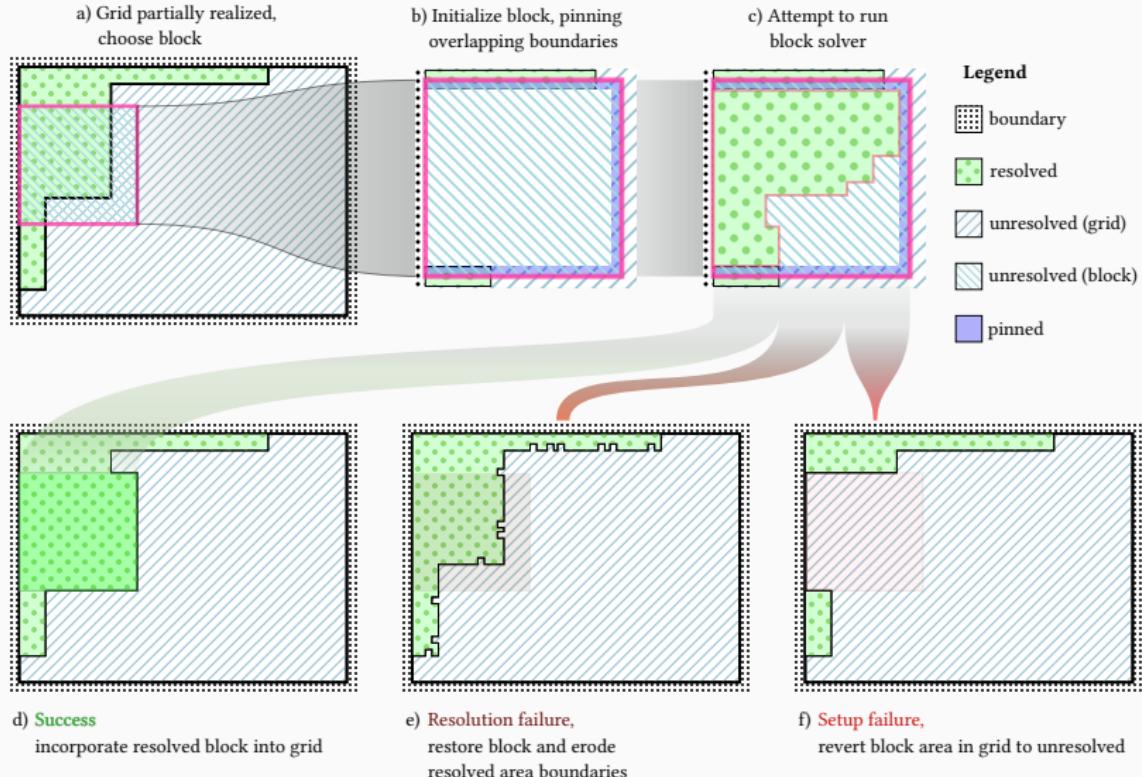
Setup failure

Failed initial
Arc Consistency

Revert area



Algorithm



Algorithm

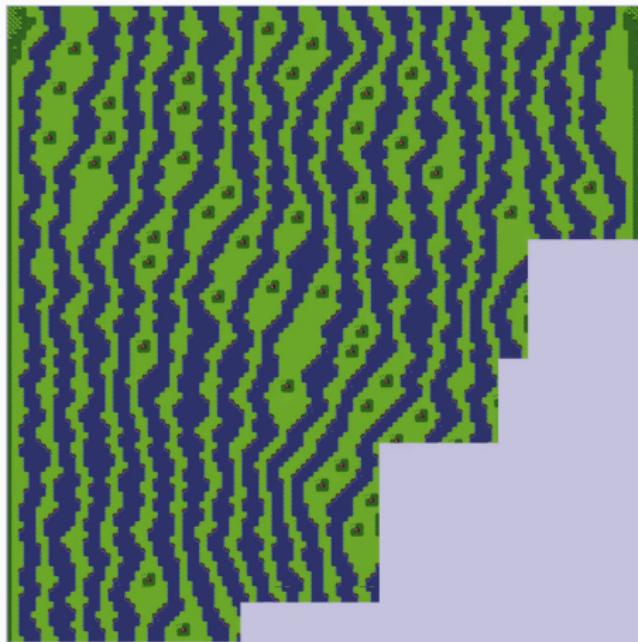
Pill Mortal Tile Set

Algorithm

ThKaspar's *Forest Micro* Tile Set

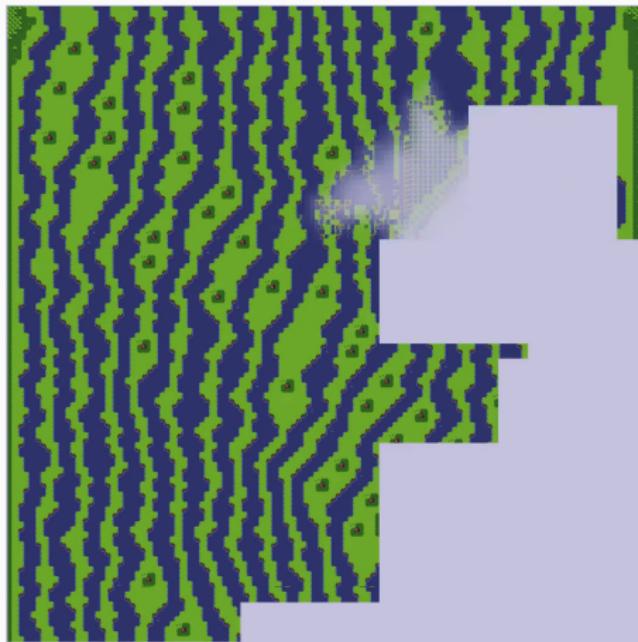
Algorithm

Revert Block



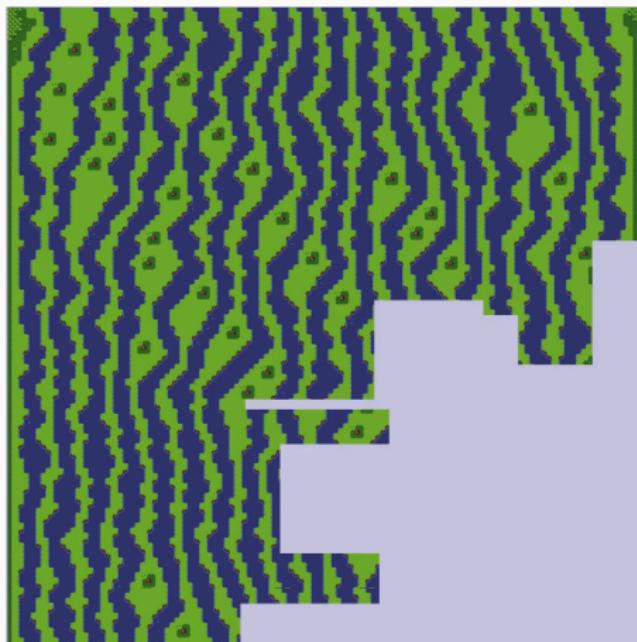
Algorithm

Revert Block



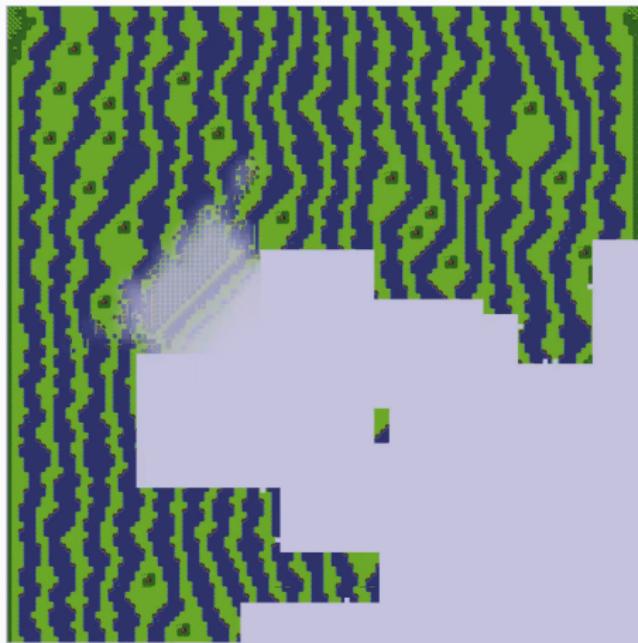
Algorithm

Erode Boundary



Algorithm

Erode Boundary



Highlighted Runs

LUNARSIGNAL's *Overhead Action*
RPG Overworld Tile Set (x10)

Highlighted Runs

0x72's *Two Bit Micro Metroidvania* Tile Set (x10)

Highlighted Runs

Kingel's *Minirogue* Tile Set (x10)

Conclusion

Punch Out Model Synthesis (POMS) as an alternative when large models with minimal setup restrictions are wanted and resource limits are a concern.

Conclusion

CBTGs are good at maintaining local consistency but are bad at resolving global constraints

Weak global constraints (path connections, etc.) confound POMS (and other CBTG algorithms)

fin

<https://zzyzek.github.io>

<https://github.com/zzyzek/PunchOutModelSynthesis>

<https://zzyzek.github.io/PunchOutModelSynthesisWebDemo/>

Auxiliary Slides

- Bitter lesson includes learning *and* search
- Tradeoff between resources used to learn vs. resources used for runtime search
- "Parables of the Power of Planning in AI" by Noam Brown (<https://www.youtube.com/watch?v=eaAonE58sLU>)

Auxiliary Slides

- Salad
- Oatmeal
- Global Cohesion/(weak) Global Constraints

Auxiliary Slides

Automatic Tile Generation

<https://zzyzek.github.io/TileRuleHighlighter/>

Rule Graph (Forest Micro)

Rule Highlighter

