Punch Out Model Synthesis

A Stochastic Algorithm for Constraint Based Tiling Generation

Zzyv Zzyzek

November 19th, 2024

Punch Out Model Synthesis

A Constraint Based Tiling
Generation algorithm

- Contradiction resilience
- Works on large grids
- Minimal setup requirements
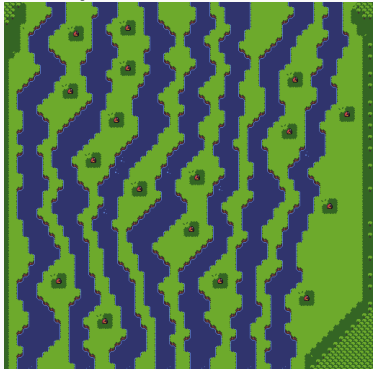
Pill Mortal Tile Set
64x64 cells, 190 tile count

Constraint Based Tiling Generation (*CBTG*) Problem

Find a valid grid realization

A *realization* is a single *tile* placement at each *cell* respecting *constraints*.
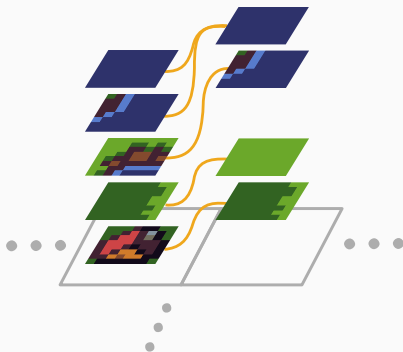
(*Cells* hold array of *tiles*)
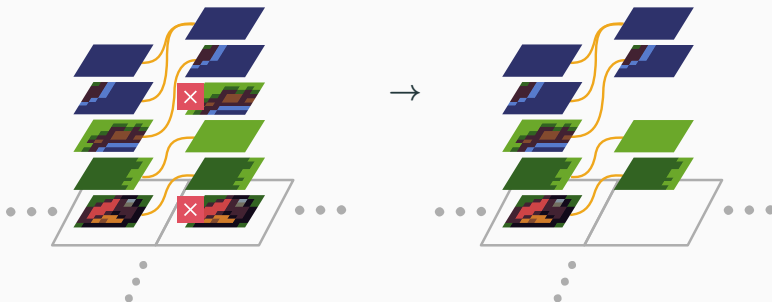
Example

64x64 cells, 159 tile count

Arc Consistency: all *tiles* in every *cell* have at least one valid neighbor in each direction

Basis for *Constraint Propagation* algorithm by repeatedly removing tiles without valid neighbors after tile choice

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

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

WFC : *Wave Function Collapse* (*Gumin*)
BMS : *Breakout Model Synthesis* (*Hoetzlein*)
MMS : *Modify in Blocks Model Synthesis* (*Merrell*)
POMS : *Punch Out Model Synthesis*

Tile Arc Consistent Correlation Length (*TACCL*) (Hoetzlein)

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

Finite correlation $\rightarrow$ independent regions

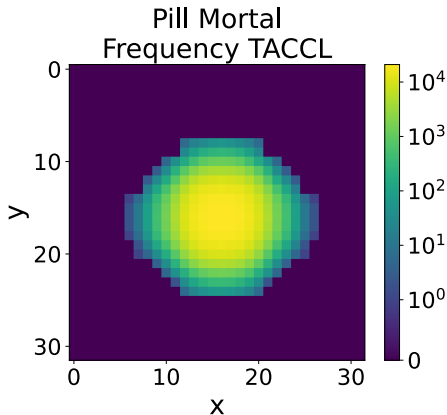*Tile Arc Consistent Correlation Length* (*TACCL*)

TACCL

Isolated block, for all tiles:

- Fix tile at the center
- Take bounding box of constraint propagation
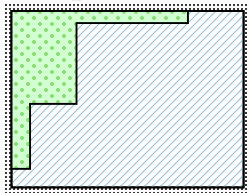
Heuristic for correlation length POMS block $\geq$ TACCL
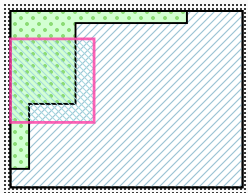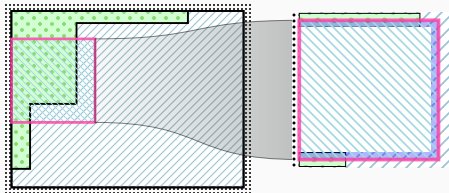
Example

Grid partially realized

cells fully realized or
marked indeterminate

Choose block
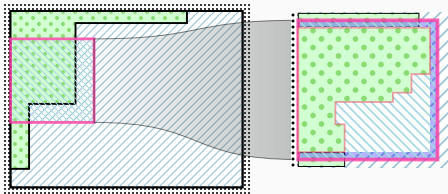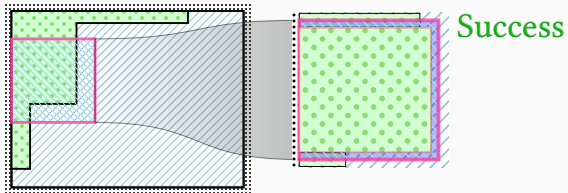
Initialize block
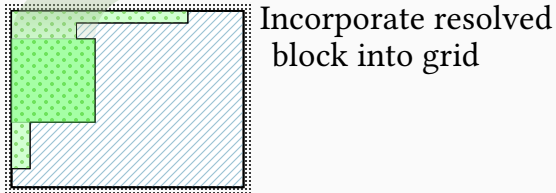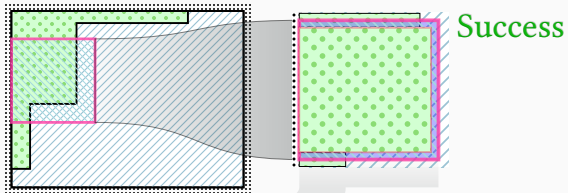 Pin boundaries
 Revert interior
 (Apply any restrictions)

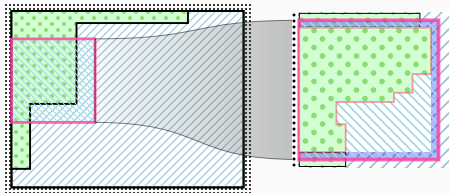Attempt to solve

block level solver

maintain full AC

Success

Success

Incorporate resolved
block into grid

Resolution failure

Resolution failure
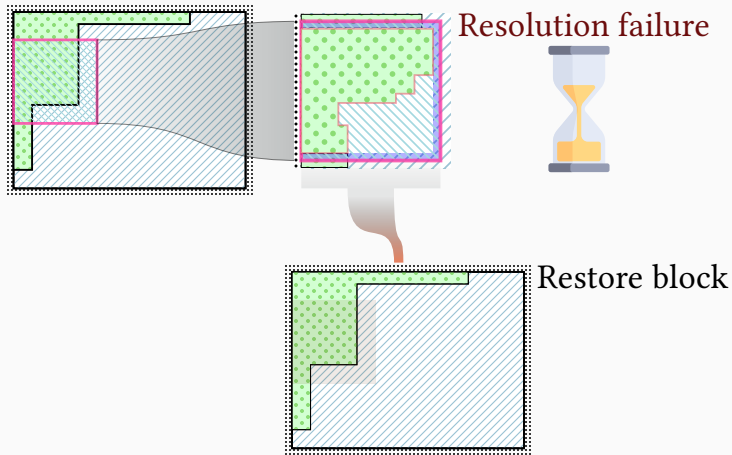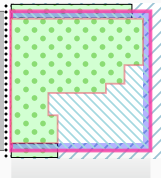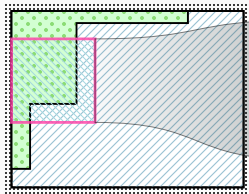
Restore block

Resolution failure
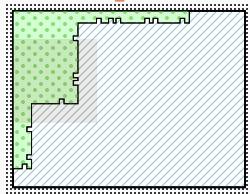
Restore block
Erode boundary

Setup failure

Failed initial
Arc Consistency
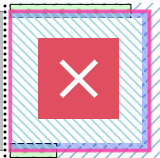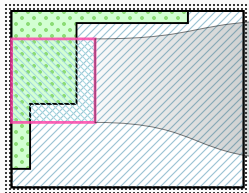
Setup failure

Failed initial
Arc Consistency

Revert area

Algorithm

a) Grid partially realized, choose block

b) Initialize block, pinning overlapping boundaries

c) Attempt to run block solver

Legend

⬚ boundary
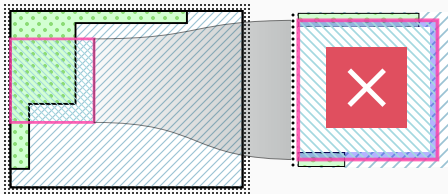
🟩 resolved

▨ unresolved (grid)

▨ unresolved (block)

🟦 pinned

d) Success
incorporate resolved block into grid

e) Resolution failure,
restore block and erode
resolved area boundaries

f) Setup failure,
revert block area in grid to unresolved
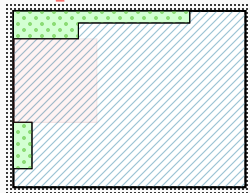
Pill Mortal
64x64 cells, 190 tiles
30x30 block

Algorithm
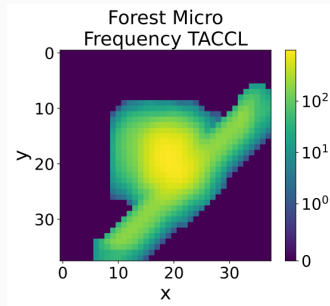
Forest Micro (ThKaspar)
128x128 cells, 159 tiles
48x48 block

Reversion and Erosion
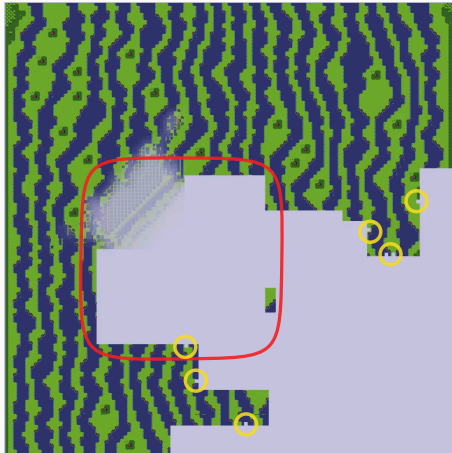
Punch Out Model Synthesis (POMS) is an alternative when:

- Grid is large
- Resource are limited
- Minimal setup requirements are needed/desired

https://zzyzek.github.io

https://github.com/zzyzek/PunchOutModelSynthesis

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

Thanks!

Automatic Tile Generation

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

Rule Graph (Forest Micro)   **Rule Highlighter**
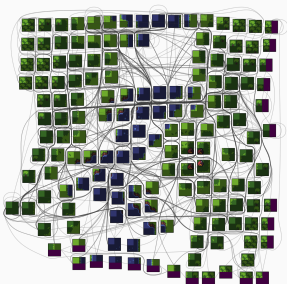
LUNARSIGNAL's *Overhead Action RPG Overworld* Tile Set (x10)

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

Kingel's *Minirogue* Tile Set (x10)

Wave Function Collapse (WFC) (Gumin)

- Block level solver (maintain AC for the block)
- One-shot solver, give up first contradiction
- Min. entropy heuristic to fix tiles

Breakout Model Synthesis (BMS) (Hoetzlein)

- Block level solver (maintain AC for the block)
- Stochastic backtracking, revert small region around contradiction point to indeterminate

Modify in Blocks Model Synthesis (MMS) (Merrell)

- Grid level solver (maintain summary info for grid, AC only for block)
- Try to resolve grid, re-incorporate if resolution successful
- Never fails (block step consistency)
- Needs fully resolved initial state
- Misses solution space if features larger than block and not already in grid (non-ergodic)

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

Other Problems

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

CBTG algorithms are good at maintaining local consistency but are bad at resolving global constraints

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

Sometimes global constraints are weak enough to be overcome by solving local constraints

Potential Future Work

- Spectral Graph Decomposition methods for automatic biome detection
- AC4 speedups via templates
- Weak global constraints