ParametrizedMazeGen

•••

Corentin Moiny
University of Montreal

IFT 6253

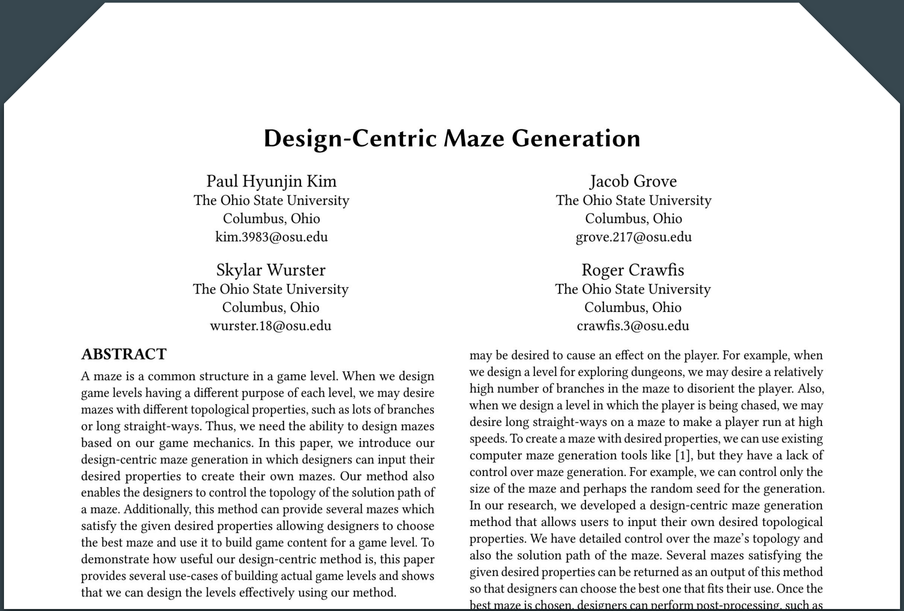
Plan

- In Context
- Motivation & Goal
- Solution
  - Development Tools
  - Architecture
  - Metamodel
  - Transformations
  - Class Diagram
  - Maze Cells
- Demo
- Future Works
- Conclusion

In Context

The Paper

- ❖ Give design aspect to a Maze
- ❖ 15 types of maze cells
- ❖ Properties on each cells
- ❖ Generation is in two parts



Design-Centric Maze Generation

Paul Hyunjin Kim
The Ohio State University
Columbus, Ohio
kim.3983@osu.edu

Jacob Grove
The Ohio State University
Columbus, Ohio
grove.217@osu.edu

Skylar Wurster
The Ohio State University
Columbus, Ohio
wurster.18@osu.edu

Roger Crawfis
The Ohio State University
Columbus, Ohio
crawfis.3@osu.edu

ABSTRACT

A maze is a common structure in a game level. When we design game levels having a different purpose of each level, we may desire mazes with different topological properties, such as lots of branches or long straight-ways. Thus, we need the ability to design mazes based on our game mechanics. In this paper, we introduce our design-centric maze generation in which designers can input their desired properties to create their own mazes. Our method also enables the designers to control the topology of the solution path of a maze. Additionally, this method can provide several mazes which satisfy the given desired properties allowing designers to choose the best maze and use it to build game content for a game level. To demonstrate how useful our design-centric method is, this paper provides several use-cases of building actual game levels and shows that we can design the levels effectively using our method.

may be desired to cause an effect on the player. For example, when we design a level for exploring dungeons, we may desire a relatively high number of branches in the maze to disorient the player. Also, when we design a level in which the player is being chased, we may desire long straight-ways on a maze to make a player run at high speeds. To create a maze with desired properties, we can use existing computer maze generation tools like [1], but they have a lack of control over maze generation. For example, we can control only the size of the maze and perhaps the random seed for the generation. In our research, we developed a design-centric maze generation method that allows users to input their own desired topological properties. We have detailed control over the maze's topology and also the solution path of the maze. Several mazes satisfying the given desired properties can be returned as an output of this method so that designers can choose the best one that fits their use. Once the best maze is chosen, designers can perform post-processing, such as

[1] - Paul Hyunjin Kim, Jacob Grove, Skylar Wurster, and Roger Crawfis. 2019. Design-centric maze generation. In Proceedings of the 14th International Conference on the Foundations of Digital Games (FDG '19), Association for Computing Machinery, New York, NY, USA, 1–9. DOI:https://doi.org/10.1145/3337722.3341854

Motivation & Goal

Motivation

- Straight forward generation
- Visual generation
- User friendly

Goals

- Define a DSL
- Produce model instances
- Define a M2T
- Output a Json file
- Implement a maze generator

Solution

Development Tools

MDE:

epsilon

emf

ECLIPSE MODELING FRAMEWORK

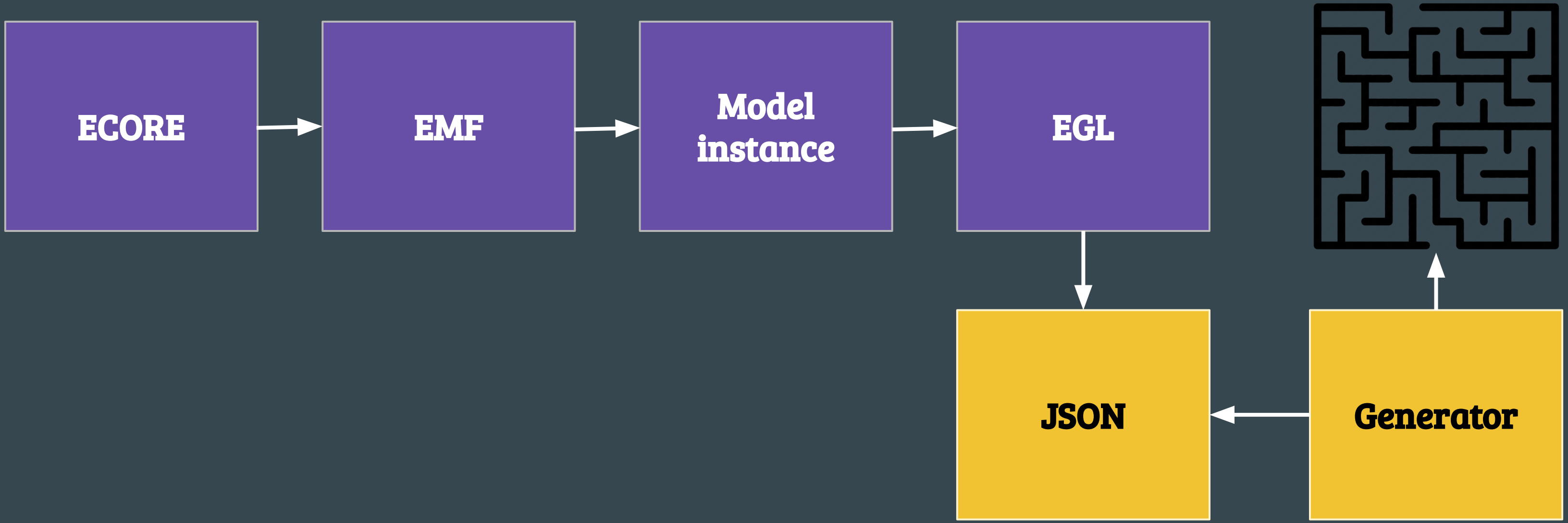
EGL

ECORE

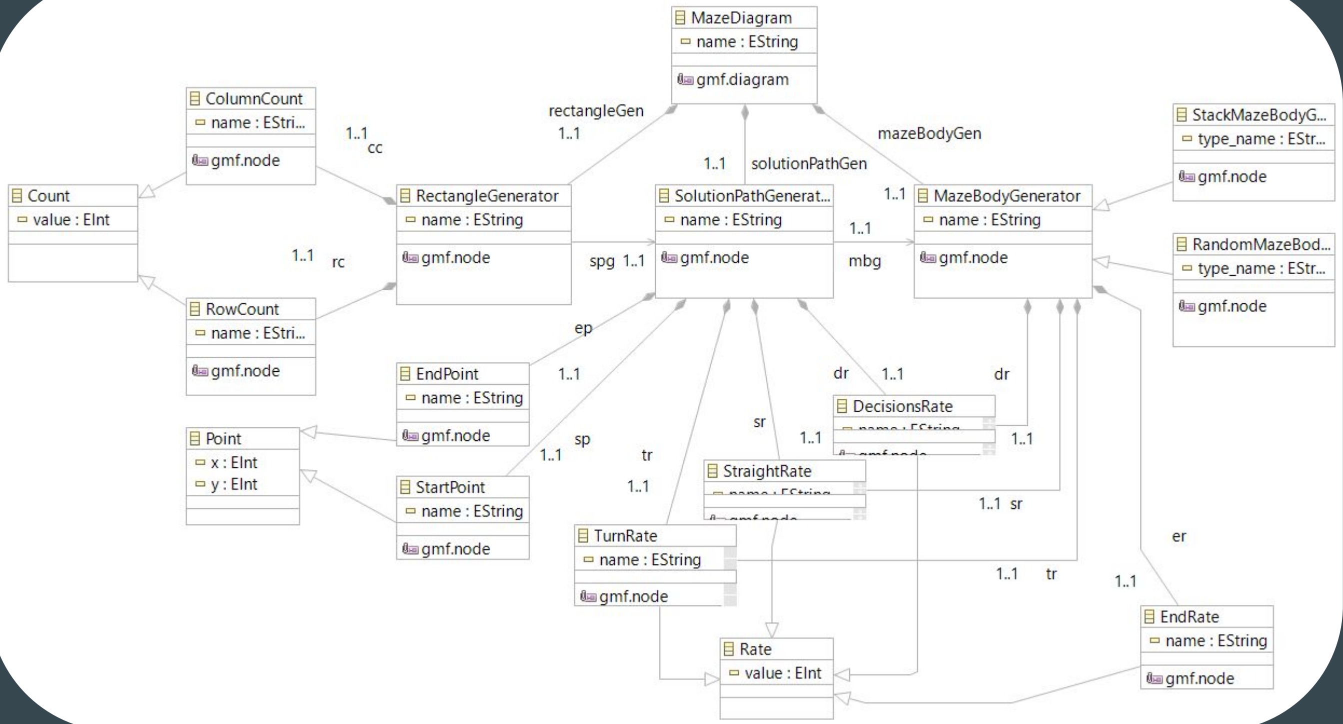
Generator:

NumPy

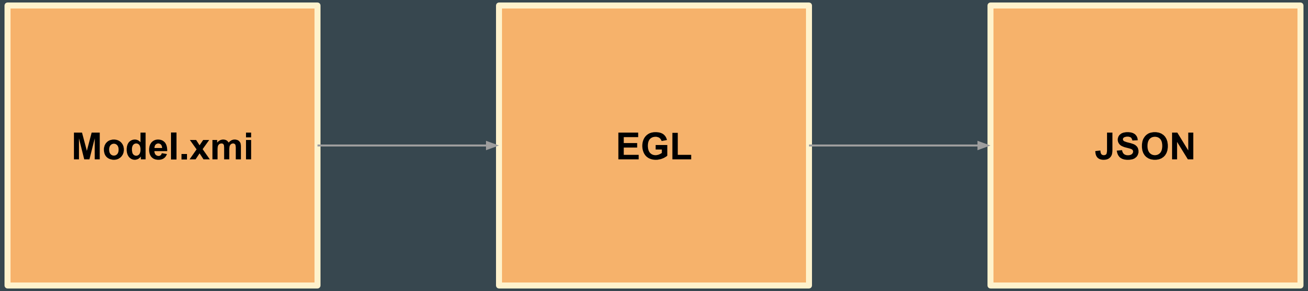
Architecture



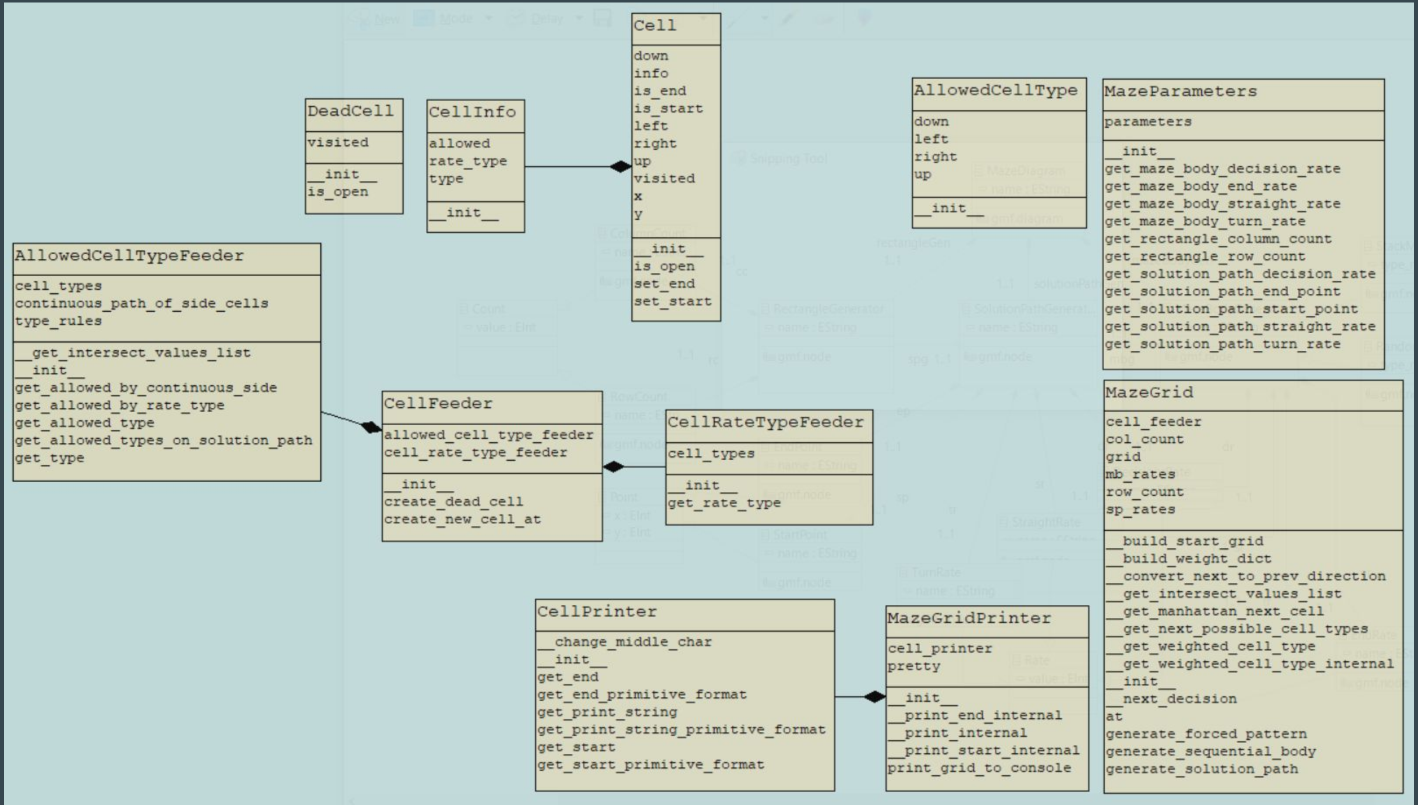
Metamodel


MazeDiagram
name : EString
gmf.diagram
rectangleGen
1..1
solutionPathGen
1..1
mazeBodyGen
ColumnCount
name : EStri...
gmf.node
1..1
cc
Count
value : EInt
RectangleGenerator
name : EString
gmf.node
1..1
rc
RowCount
name : EStri...
gmf.node
SolutionPathGenerat...
name : EString
gmf.node
spg 1..1
1..1
MazeBodyGenerator
name : EString
gmf.node
1..1
mbg
StackMazeBodyG...
type_name : EStr...
gmf.node
RandomMazeBod...
type_name : EStr...
gmf.node
ep
EndPoint
name : EString
gmf.node
1..1
Point
x : EInt
y : EInt
sp
1..1
StartPoint
name : EString
gmf.node
tr
1..1
sr
1..1
dr
1..1
dr
DecisionsRate
1..1
StraightRate
1..1 sr
TurnRate
name : EString
gmf.node
1..1
tr
er
1..1
EndRate
name : EString
gmf.node
Rate
value : EInt

Transformation



Class Diagram



Cell
down
info
is_end
is_start
left
right
up
visited
x
y
__init__
is_open
set_end
set_start

DeadCell
visited
__init__
is_open

CellInfo
allowed
rate_type
type
__init__

AllowedCellType
down
left
right
up
__init__

MazeParameters
parameters
__init__
get_maze_body_decision_rate
get_maze_body_end_rate
get_maze_body_straight_rate
get_maze_body_turn_rate
get_rectangle_column_count
get_rectangle_row_count
get_solution_path_decision_rate
get_solution_path_end_point
get_solution_path_start_point
get_solution_path_straight_rate
get_solution_path_turn_rate

AllowedCellTypeFeeder
cell_types
continuous_path_of_side_cells
type_rules
__get_intersect_values_list
__init__
get_allowed_by_continuous_side
get_allowed_by_rate_type
get_allowed_type
get_allowed_types_on_solution_path
get_type

CellFeeder
allowed_cell_type_feeder
cell_rate_type_feeder
__init__
create_dead_cell
create_new_cell_at

CellRateTypeFeeder
cell_types
__init__
get_rate_type

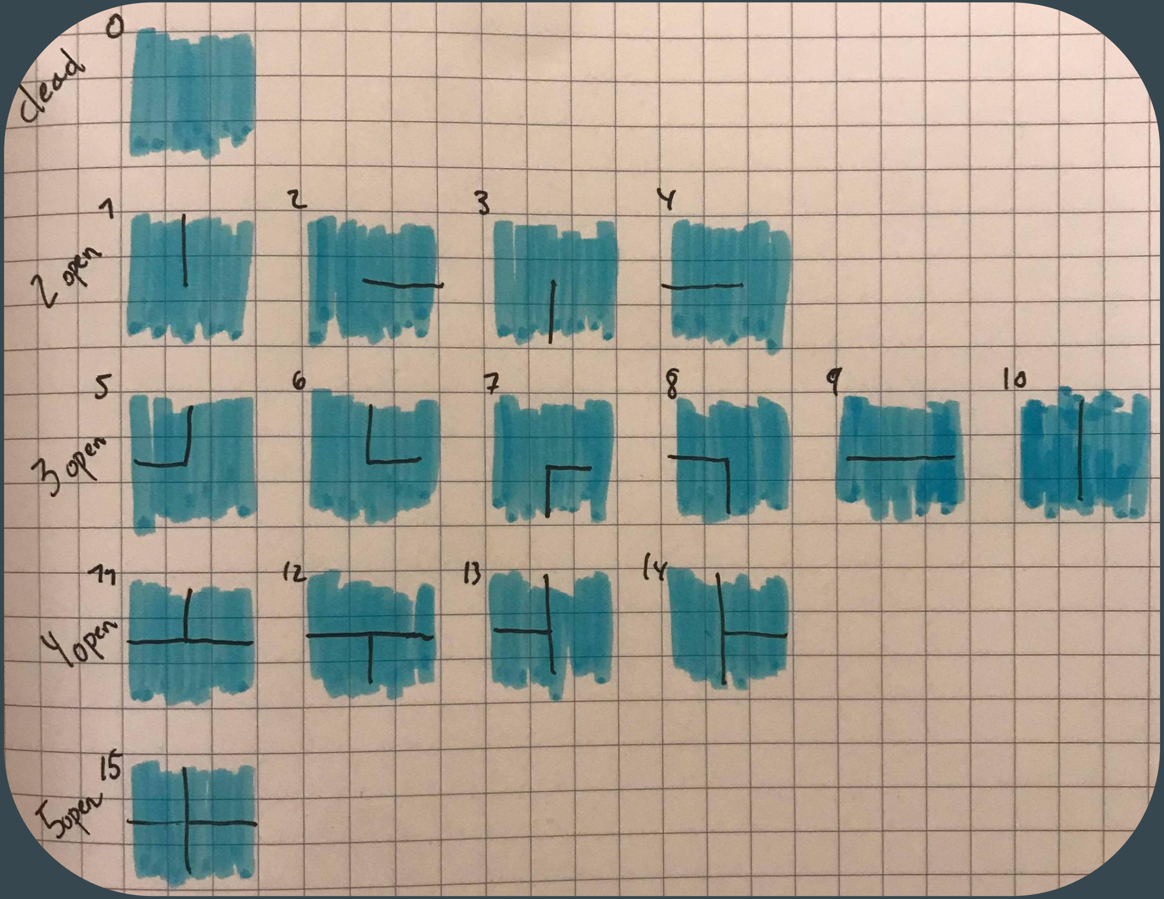
MazeGrid
cell_feeder
col_count
grid
mb_rates
row_count
sp_rates
__build_start_grid
__build_weight_dict
__convert_next_to_prev_direction
__get_intersect_values_list
__get_manhattan_next_cell
__get_next_possible_cell_types
__get_weighted_cell_type
__get_weighted_cell_type_internal
__init__
__next_decision
at
generate_forced_pattern
generate_sequential_body
generate_solution_path

CellPrinter
__change_middle_char
__init__
get_end
get_end_primitive_format
get_print_string
get_print_string_primitive_format
get_start
get_start_primitive_format

MazeGridPrinter
cell_printer
pretty
__init__
__print_end_internal
__print_internal
__print_start_internal
print_grid_to_console

Maze Cells

- 16 types of cells
- Open/Close sides
- 4 rates types

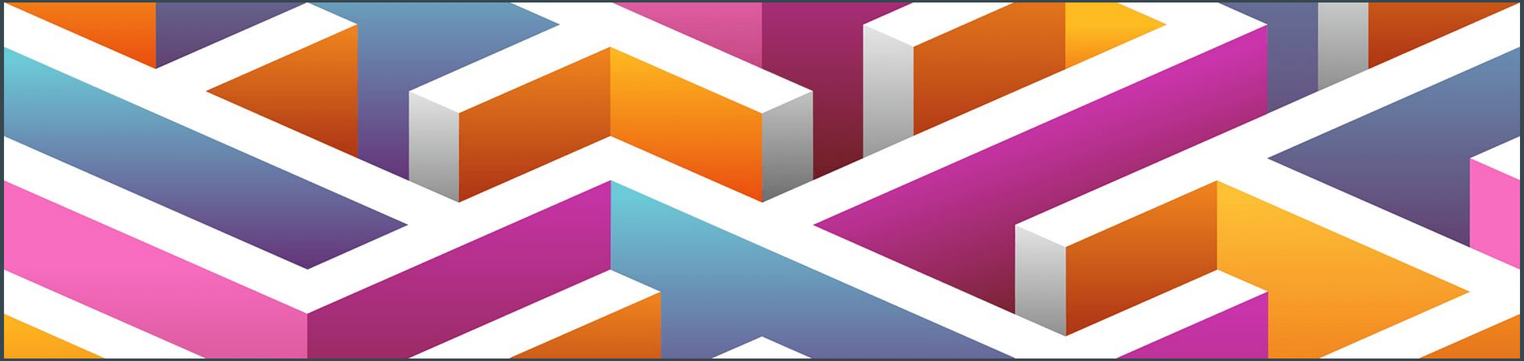


Demo

Future Works

Force Pattern Feature

- Conceptualize Maze Cells in the DSL
- Generate based on this forced pattern

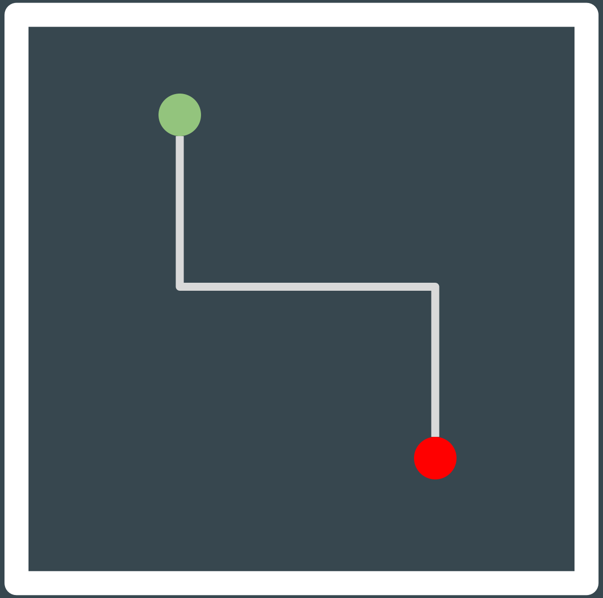


Constraints

- Start and End points within

  Column and Row counts

```
IndexError: list index out of range
```



Conclusion

To finish ...

- Github (*https://github.com/Thealoe/ParametrizedMazeGen*) - Not public yet
- Main Challenges
  - Merge Generator and DSL
  - Generator implementation