

以遊玩特徵為導向的 程序化內容生成方法

Advisor: Prof. Wen-Kai Tai

Student: Ze-Hao Wang

Agenda

- ❖ Introduction
- ❖ Related Works
 - ❖ Mission / Space framework
 - ❖ Map Sketches & Evolution of Segments
- ❖ Proposed Methodologies
 - ❖ System Architecture
 - ❖ Mission Grammars
 - ❖ Room Definitions and Instruction
- ❖ Genetic Algorithm in Segments Evolution
- ❖ Experimental Results
- ❖ Conclusions and Contributions
- ❖ Future work

Introduction

Motivation

- ❖ [會再翻成英文]
- ❖ 程序化內容自動生成 (Procedural Content Generation) 在過去就廣泛被應用於遊戲設計領域，其主要目的為增加遊戲內容的隨機性與多樣性。我們預期讓玩家在進行遊戲時能夠遵循關卡設計師的劇情脈絡外，亦能夠體驗到有意義且多樣化的遊戲關卡內容。



Research Goals

- ❖ [會再翻成英文]
- ❖ 我們針對遊戲過程中的遊玩特徵 (gameplay patterns) 進行抽象化，使用程序化生成技術產生帶有意義遊戲關卡內容，藉此消彌或降低因隨機性所產生的不穩定要素，以改善並豐富遊戲體驗。

Related Works



Mission / Space framework

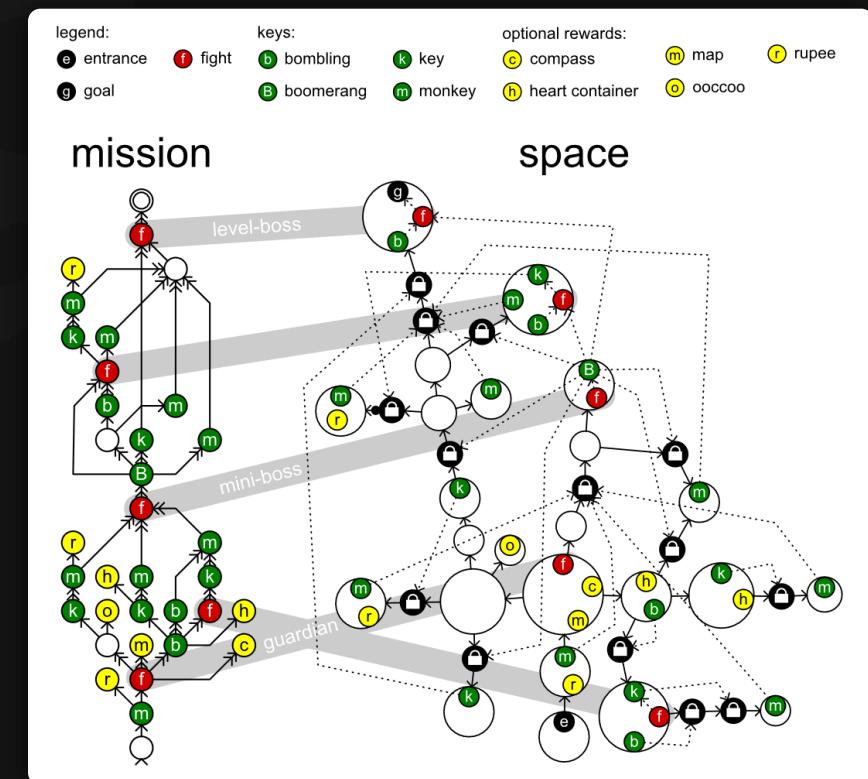


Map Sketches & Evolution of Segments

Mission / Space framework

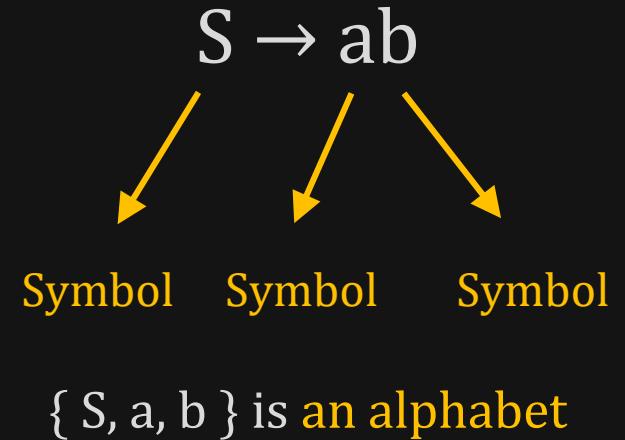
Mission/Space framework, focuses on **level design** and the **mechanics** that control player progression through a game.

- ❖ Transformational Grammars
- ❖ Mission Grammars
- ❖ Space Grammars
- ❖ Mission Graph Convert into Space Graph



Transformational Grammars

- ❖ Be consisted from **an alphabet** and **a set of rules**
 - ❖ **Alphabet**
It is a set of symbols the grammar works with.
 - ❖ **Set of rules**
It specifies what symbol can be replaced by what other symbols to form a new string.
 - ❖ **Sides and symbols of the rule**
 - ❖ **Terminals** (common in lowercase)
Symbols in the alphabet can never be replaced because there are no rules.
 - ❖ **Non-terminals** (common in uppercase)
Symbols have rules that specify their replacement.



Mission Grammars

- ❖  **Inhibitions**

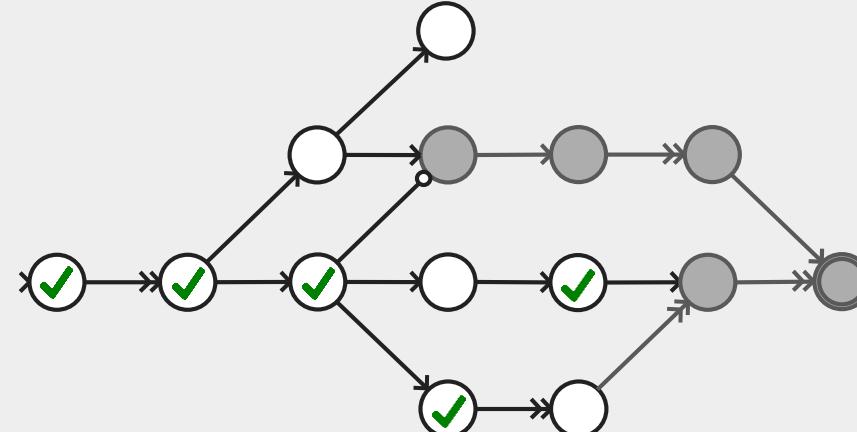
Available when at least one of the weak prerequisites is completed.

- ❖  **Strong requirement**

Available when all strong prerequisites are completed.

- ❖  **Weak requirement**

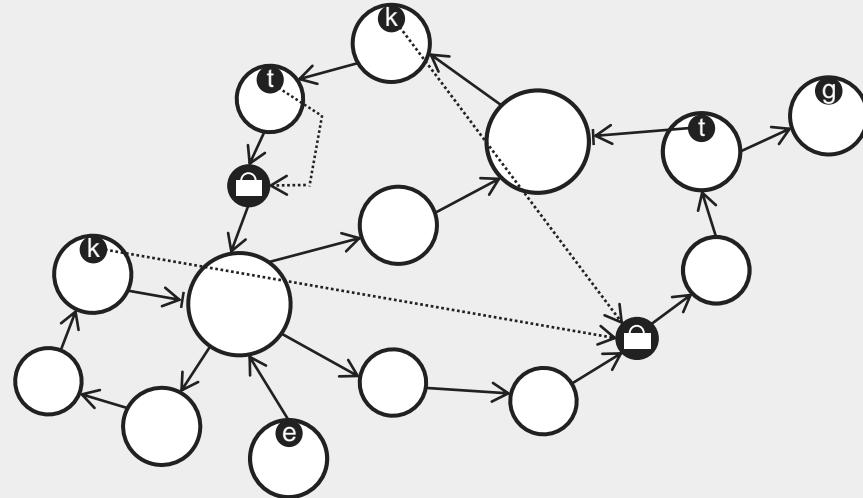
Available when at least one of the weak prerequisites is completed.



Example of mission graph

- ❖ Mission graphs represent the players' progress towards a goal not by tracking their physical location, but by tracking the tasks they must perform to finish a level.
- ❖ A mission graph is a directed graph that represents a sort of to-do list with each node representing a task that might or must be executed by the players.

Space Grammars



Mission Graph Convert into Space Graph

- ❖ The steps in the generation progress investigated in this paper in detail:

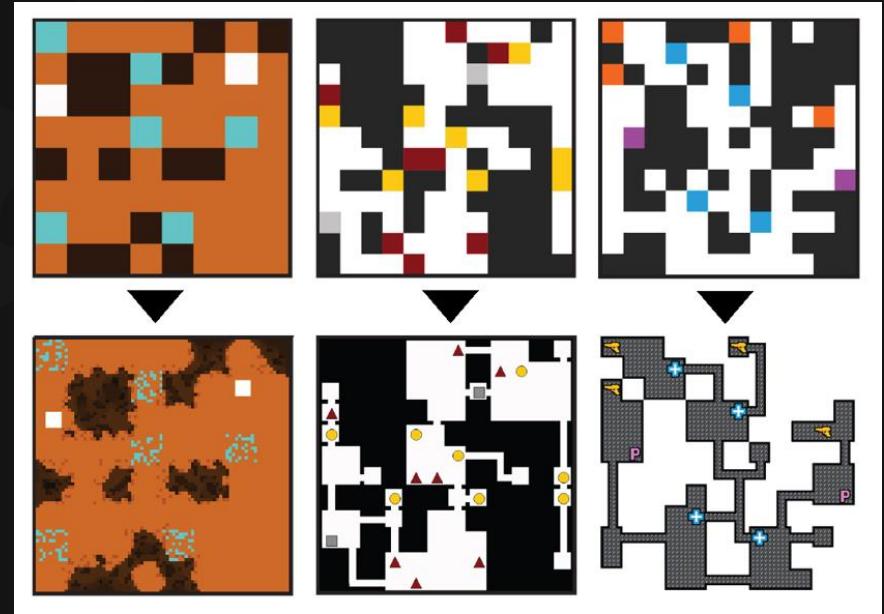


- ❖ An alternative method, does not go into detail:



Map Sketches & Evolution of Segments

- ❖ Map Sketches
- ❖ Map Sketch Evolution
- ❖ Dungeon Segments
- ❖ Dungeon Segment Evolution



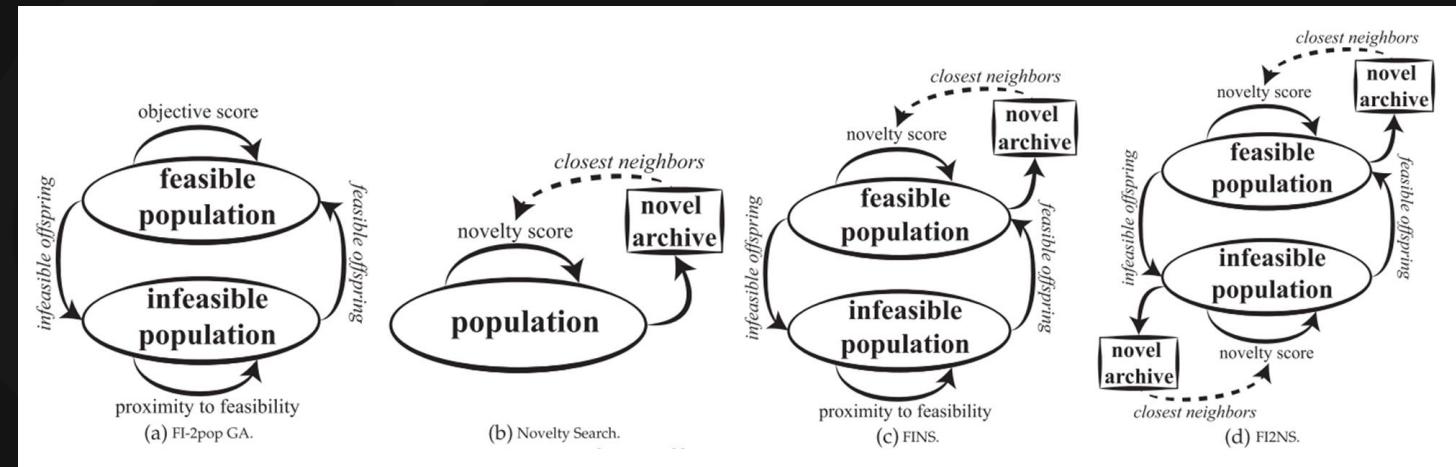
Map Sketches

Map Sketch Evolution

Dungeon Segments

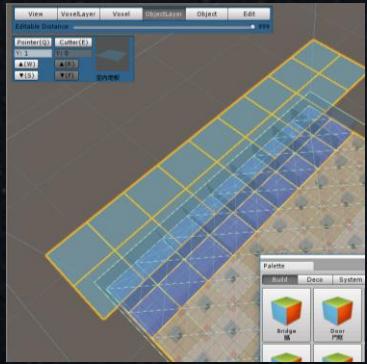
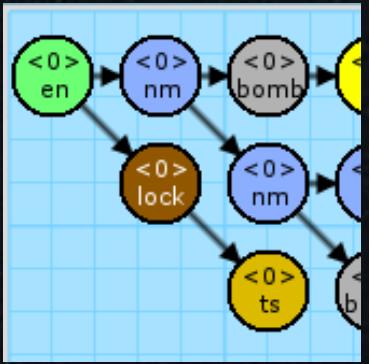
Dungeon Segment Evolution

Constrained Novelty Search



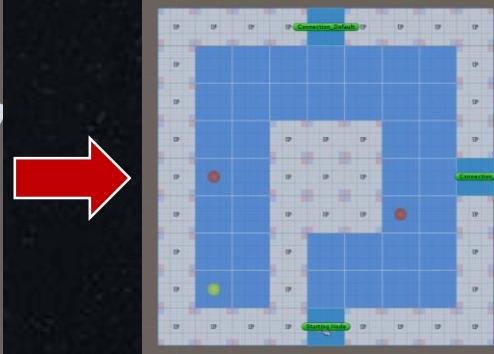
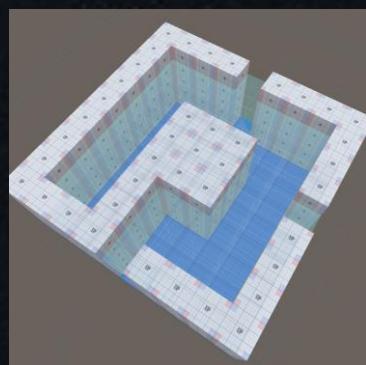
Proposed Methodologies

System Architecture

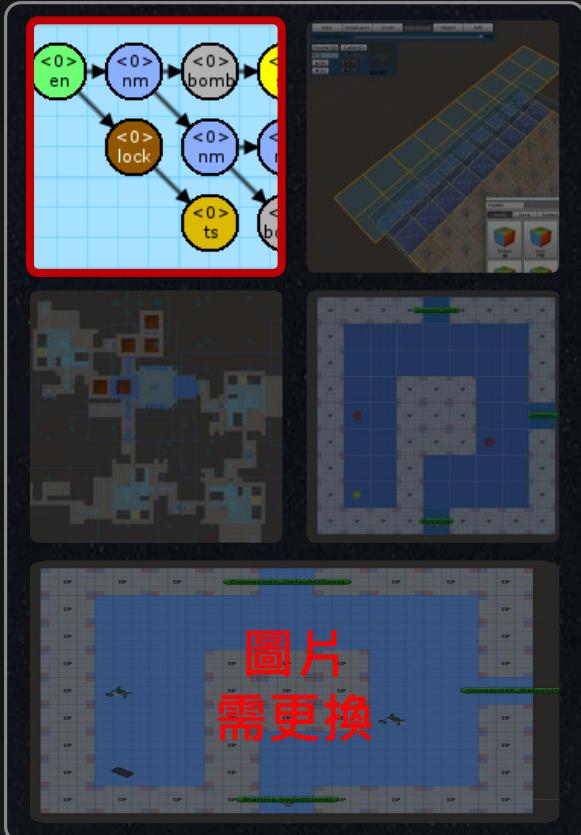


- Phase 1 -
Mission / Space framework

- Phase 2 -
Dungeon Segment Evolution

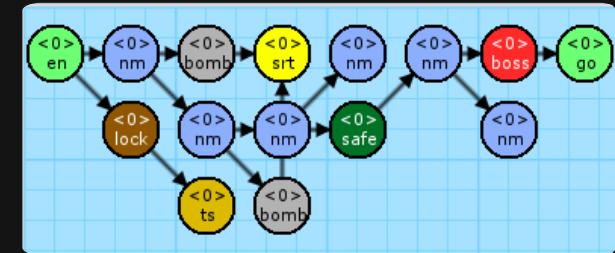


System Architecture

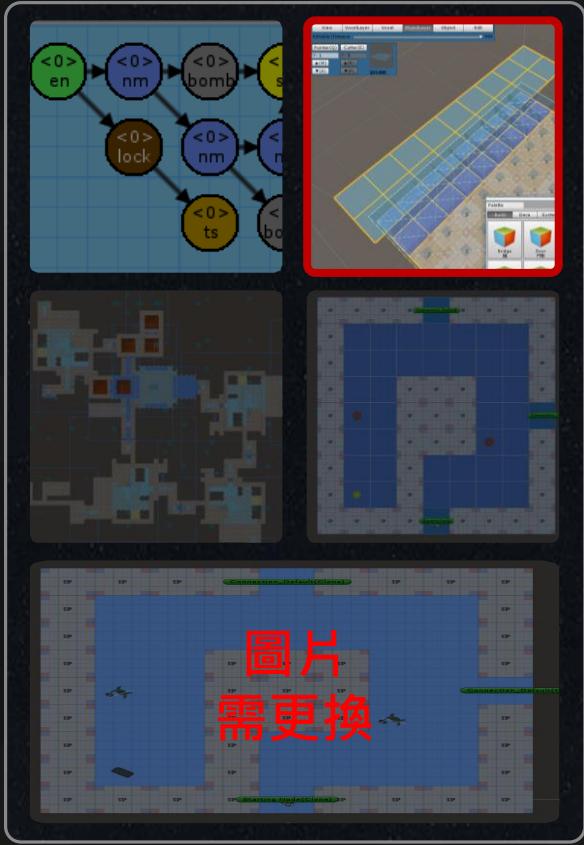


1. Create Mission Grammars

- ❖ Mission Alphabet
 - ❖ Terminal nodes & non-terminal nodes
- ❖ Mission Rules

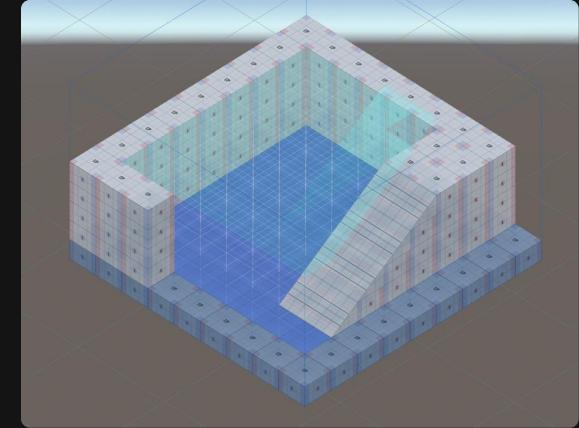


System Architecture

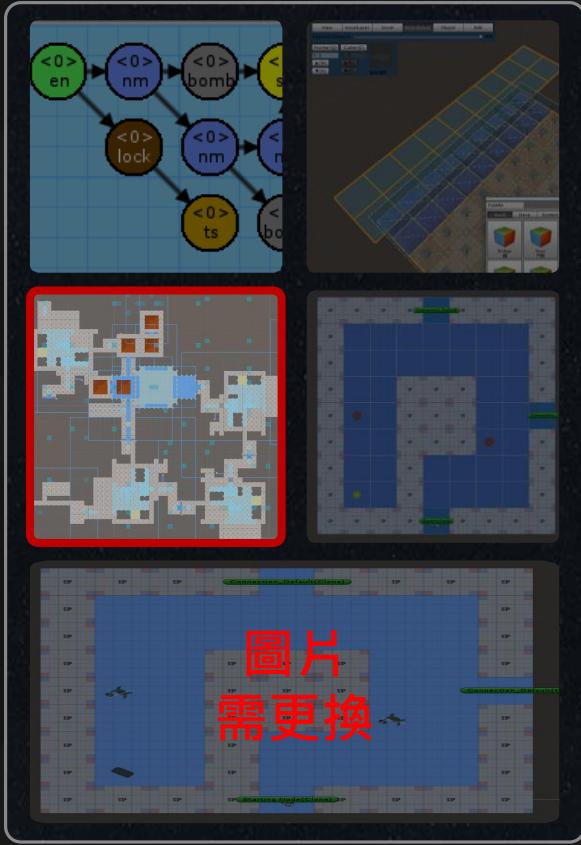


2. Create Volumes

- ❖ Volex-based volumes



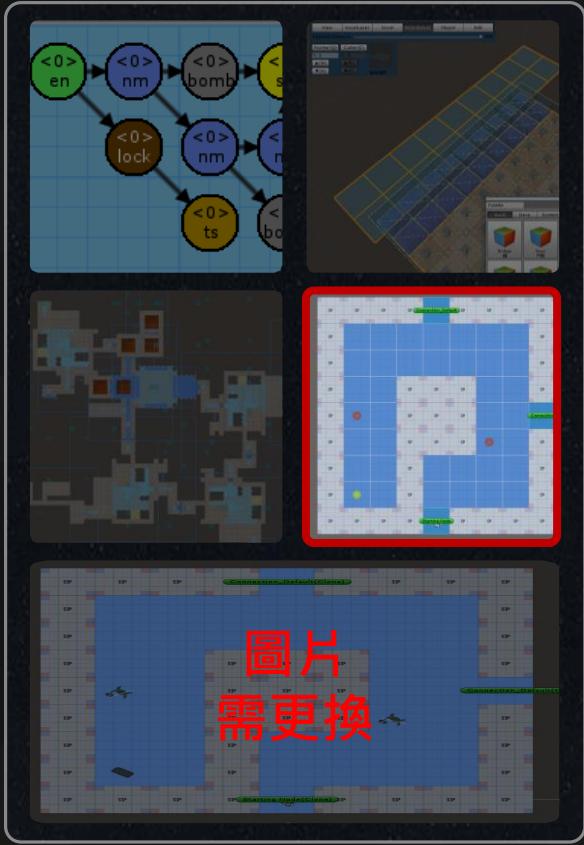
System Architecture



3. Generate the Space

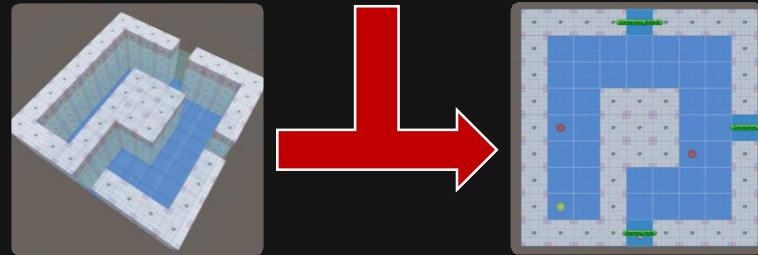
- ❖ Instruction of Rewrite System
- ❖ Replacement of Rewrite System

System Architecture

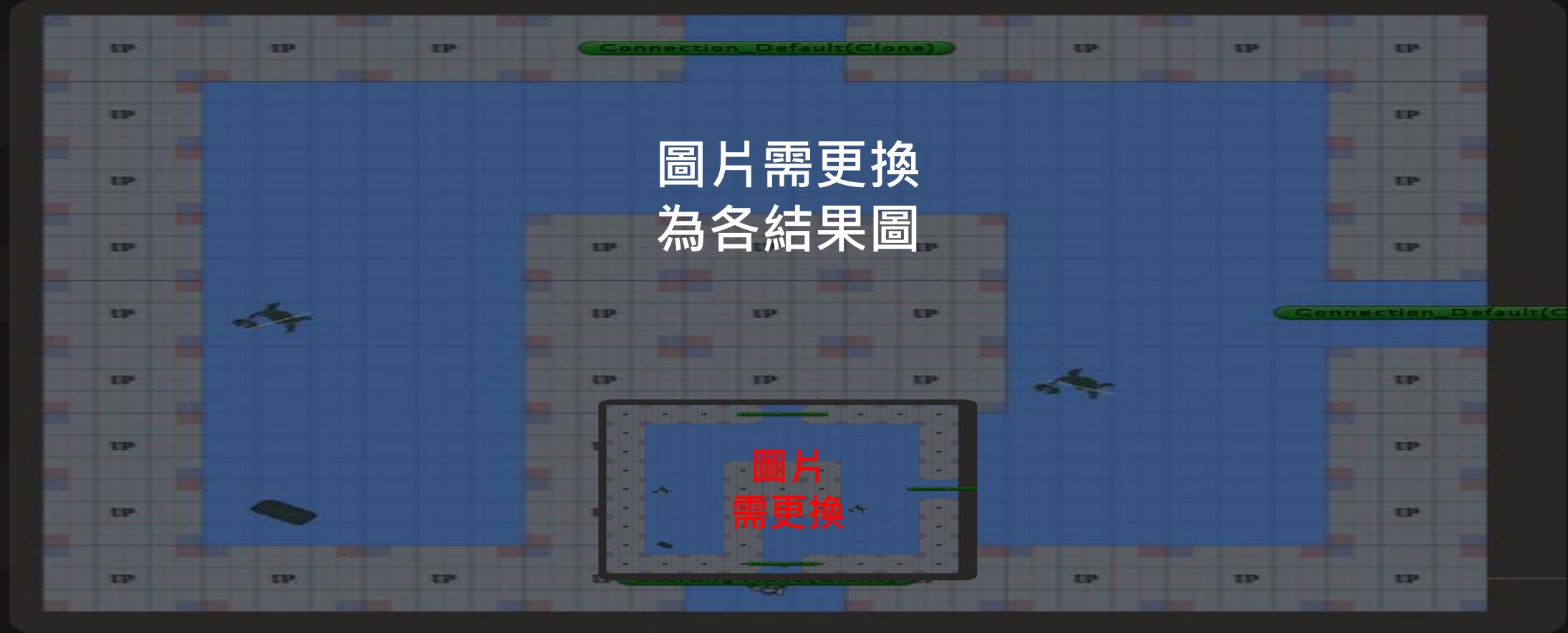


4. Generate the Game Patterns

- ❖ Genetic Algorithm for emergence objects
- ❖ Nine main metrics to design the fitness functions



System Architecture



System DEMO

影片將會放置於此

Mission Grammars

Nodes **Connections**

LIST OF NODES

- SIL
- <0> ts
- <0> S
- <0> SP

Add New Modify Delete

<0> S

Symbol Type: Non Terminal
Name: Start
Abbreviation: S
Description: Start
Outline Color: Filled Color: Text Color:

The data is up to date.

Update the changes

Current Group: Main Dungeon
Current Rules: Main Path
Rule Name: Main Path
Rule Description: Main path with multi branches.
Info: The name has been used before.
Apply

SOURCE **REPLACEMENT**

<1> S

<1> en → <3> NM → <4> NM → <5> NM → <7> NM → <2> go

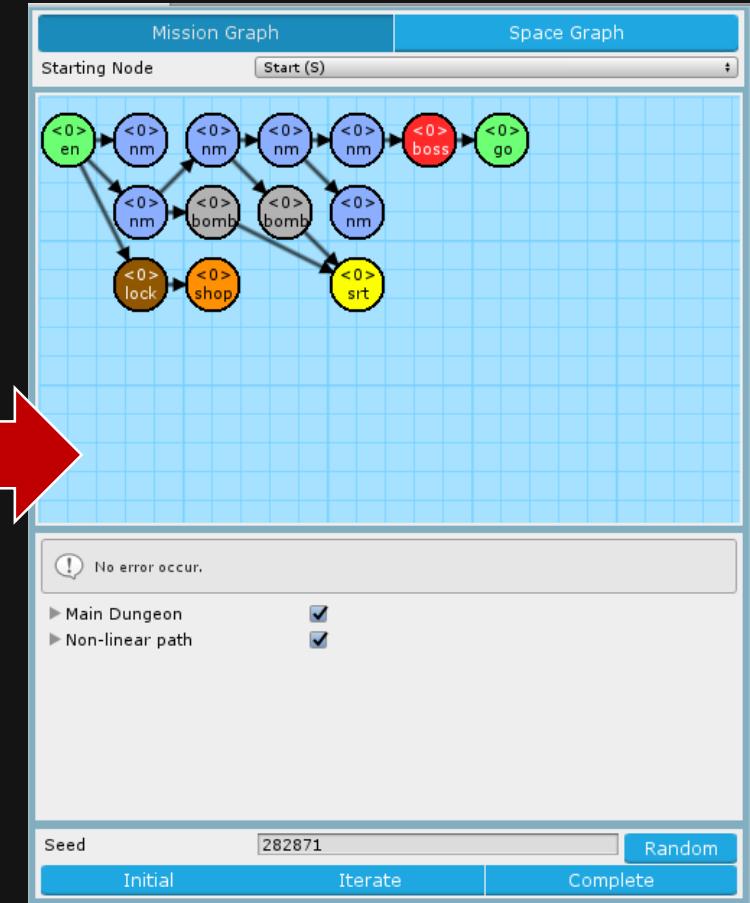
Ordering: Weight 10 Quantity limit 0 ~ 0

Add Node Add Connection Copy Delete

LIST OF NODES

- <0> boss (boss)
- <0> safe (safe)
- <0> shop (shop)

Add New Modify

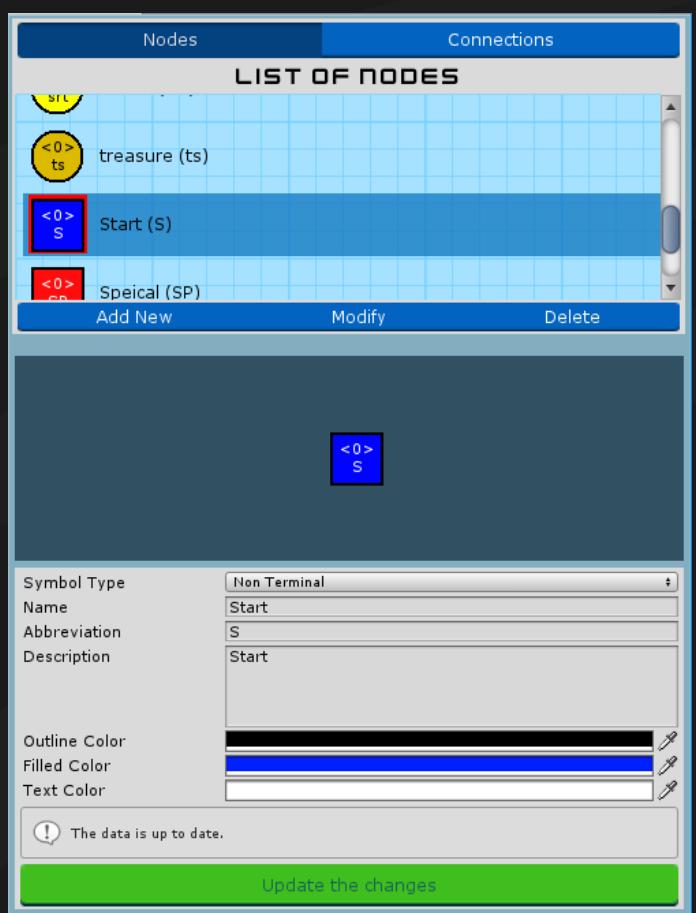


Mission Alphabet

Mission Rules

Mission Graph

Introduction of Mission Alphabet



Mission Alphabet Window

- ❖ List of symbols

- ❖ Nodes and connections
- ❖ Directly preview before selected

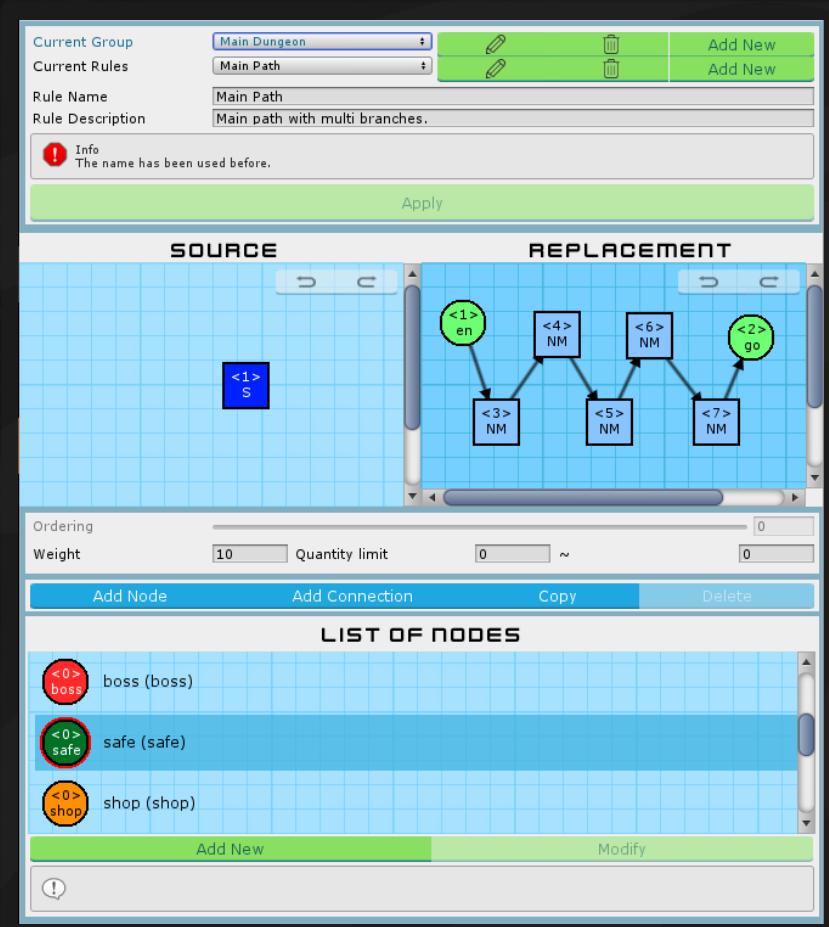
- ❖ Preview area

- ❖ Preview the symbol after editing immediately
- ❖ Submit hint and form validations

- ❖ Default and extended nodes

- ❖ Default: *None, Entrance, Goal*
- ❖ Extended: *Any*

Introduction of Mission Rules



Mission Rules Window

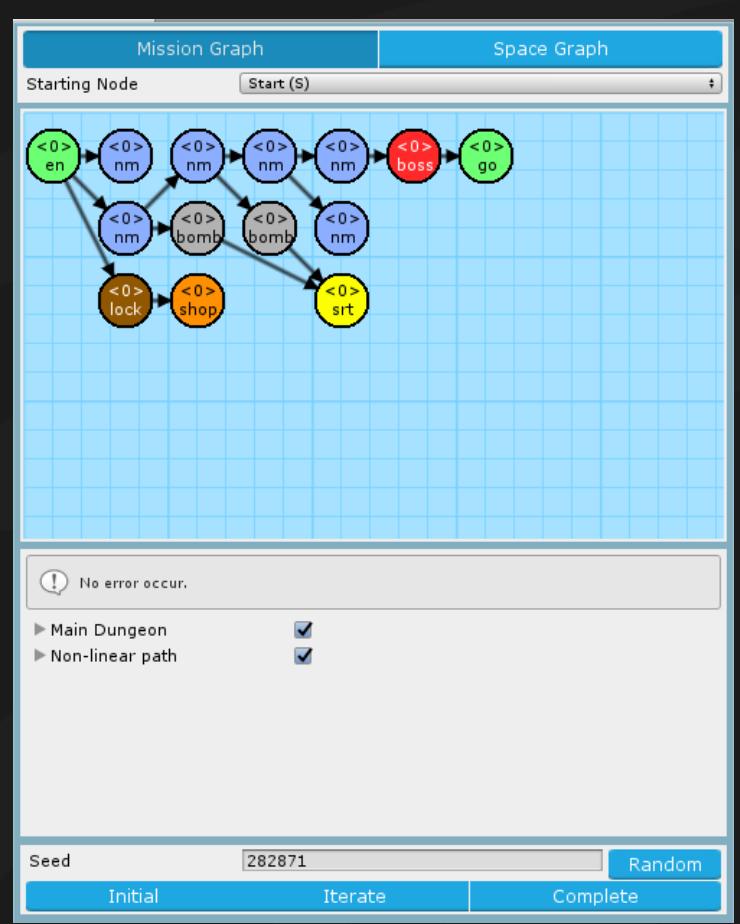
❖ Hierarchy structure

- ❖ Groups > Rules > Source & Replacement
- ❖ Friendly interface to create, delete and edit.

❖ Rule canvas

- ❖ Drag & drop to modify symbols
- ❖ Custom size of canvas
- ❖ Selected symbol highlighting
- ❖ Connections stick on nodes automatically
- ❖ Back trace the states (Redo / Undo)

Introduction of Mission Graph



Mission Graph Window

- ❖ Set starting node
 - ❖ The head of the mission graph
- ❖ Preview mission graph
- ❖ Rewrite system
 - ❖ Based on the mission grammar previous window set
 - ❖ Toggle the selectable rules
 - ❖ Find the match parts in graph with rules
 - ❖ VF Graph Isomorphism Algorithm

Usage flow

A

Create the Mission Alphabet

Usage flow

B

Design the Mission Rules

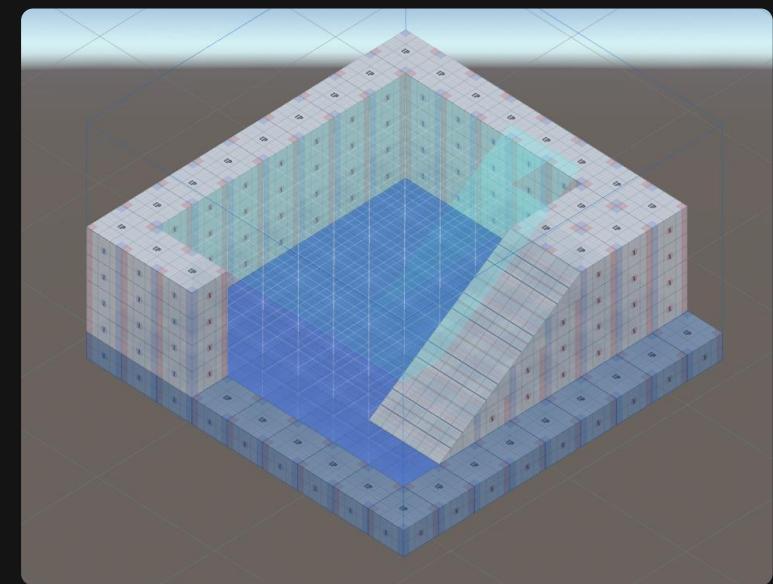
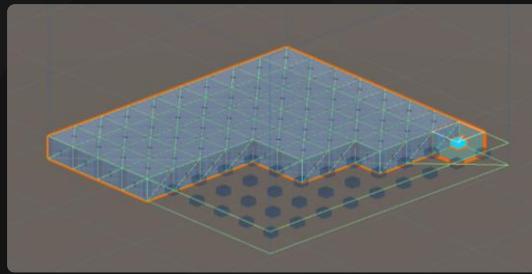
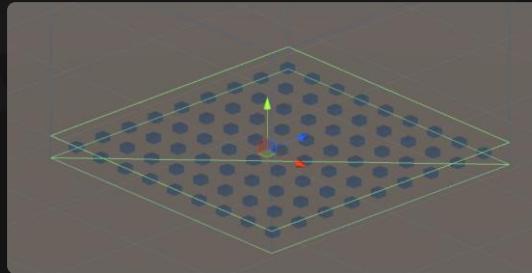
Usage flow

C

Export the Mission Graph

Room Definitions and Instruction

- ❖ Editor
 - ❖ Voxel-based units
 - ❖ Markers I - Decorations
 - ❖ Markers II - Connections
- ❖ Rewrite System
 - ❖ Instruction
 - ❖ Replacement



Voxel-based units in Editor

- ❖ Hierarchy
 - ❖ Level
A set of volumes. Express a complete game level.
 - ❖ Volume
Volume has
 - ❖ Chunk
 - ❖ Voxel

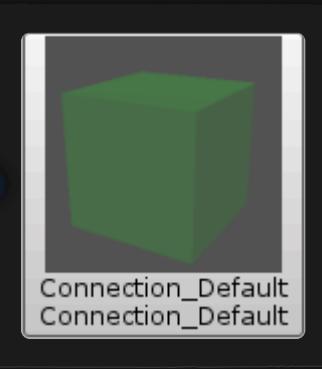
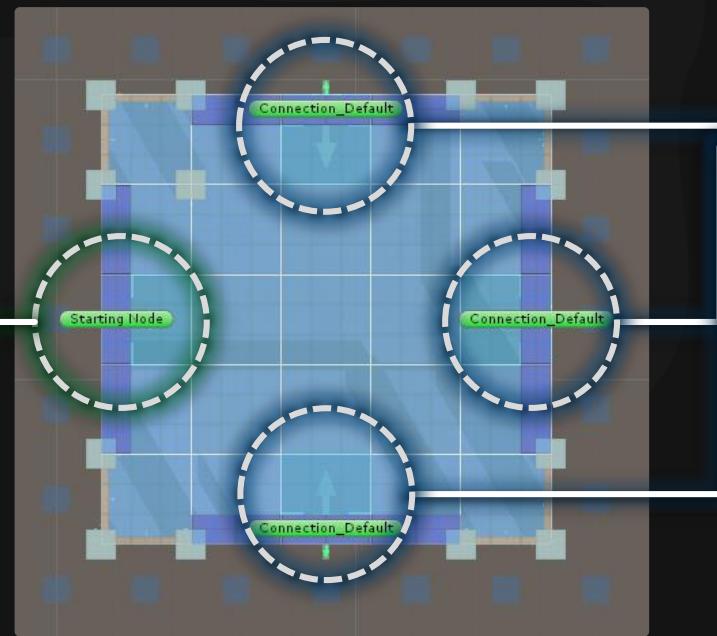
Decorations in Editor

Connections in Editor



Starting node

- ❖ Zero or one
- ❖ Inner direction



Connections

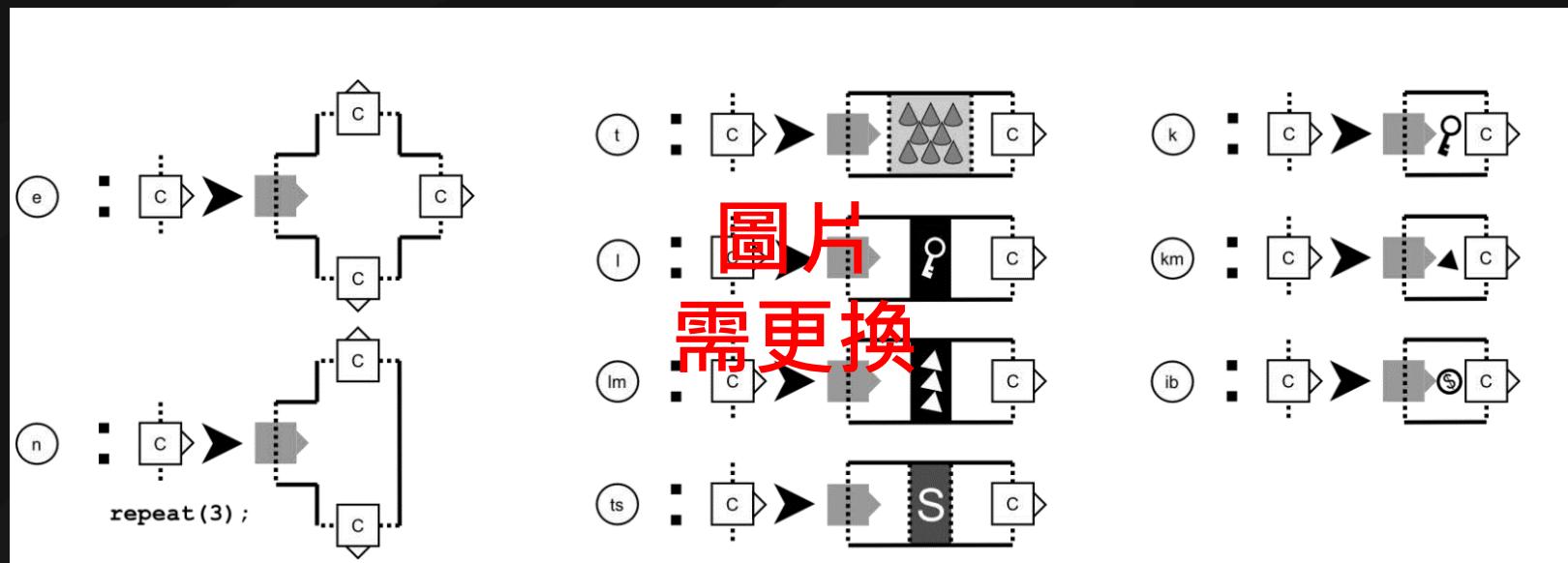
- ❖ Any amount
- ❖ Outer direction

- ❖ The **connection** of “A volume” will stick with the **starting node** of “B volume”, according to their direction of arrow. If doesn’t exist any stating node, will pick one connection randomly.

Instruction of Rewrite System

❖ Building Instructions

- ❖ Each rule in the shape grammar was associated with a **terminal symbol** form in the mission grammar.
- ❖ Look for rules that implement that symbol, selects one at random based on their **relative weight**.

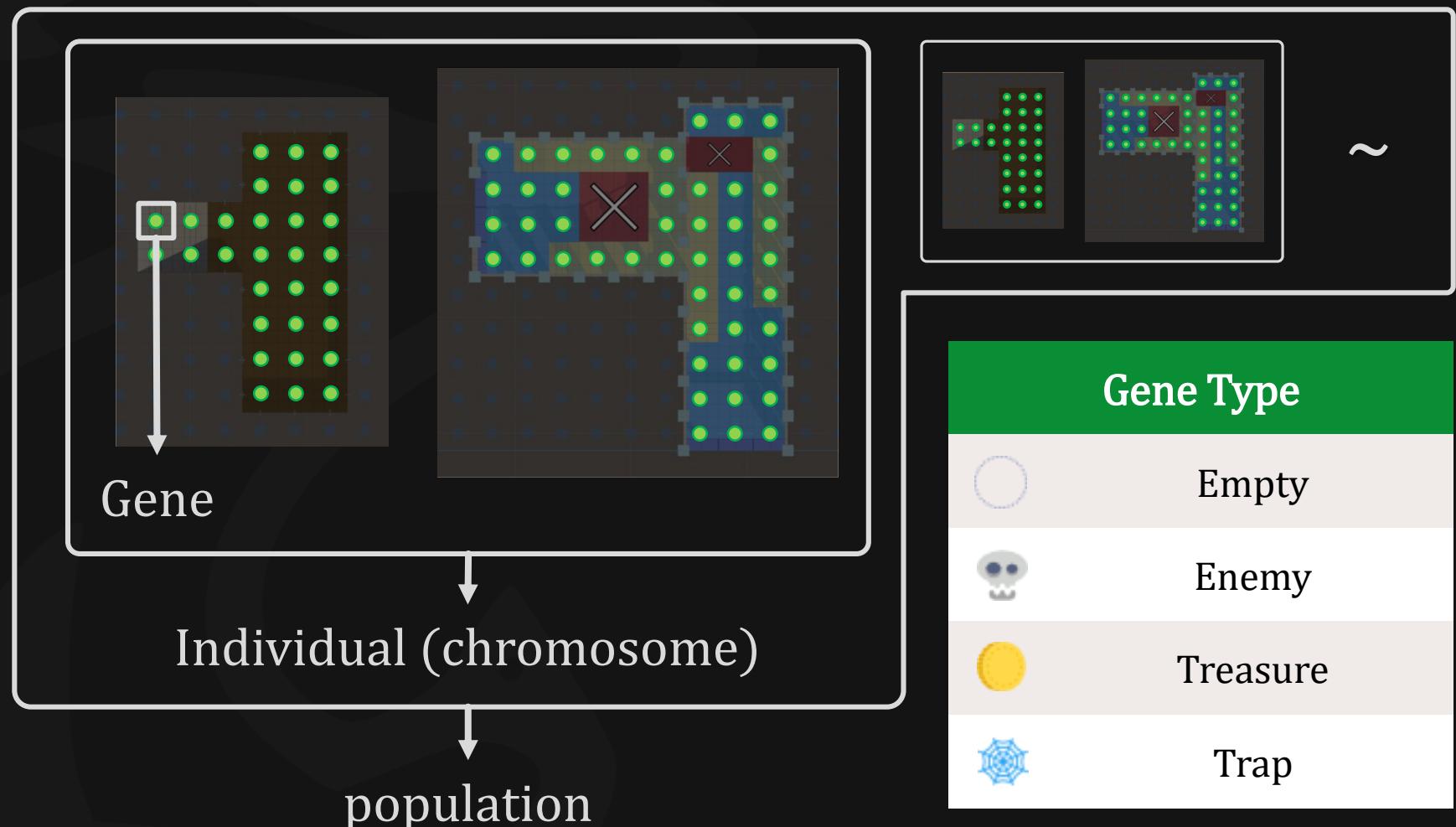


Replacement of Rewrite System

Genetic Algorithm in Segments Evolution

- ❖ Genetic Algorithm
 - ❖ Algorithm flow
 - ❖ Gene
 - ❖ Crossover
 - ❖ Mutation
- ❖ Nine fitness functions
 - ❖ Neglected
 - ❖ Block
 - ❖ Intercept
 - ❖ Patrol
 - ❖ Guard
 - ❖ Dominated
 - ❖ Support
 - ❖ Cover
 - ❖ Trap

Gene



Crossover

Mutation

Nine fitness functions – 1st Neglected

Nine fitness functions – 2nd Block

Nine fitness functions – 3rd Intercept

Nine fitness functions – 4th Patrol

Nine fitness functions – 5th Guard

Nine fitness functions – 6th Dominated

Nine fitness functions – 7th Support

Nine fitness functions – 8th Cover

Nine fitness functions – 9th Trap

Experimental Results

Conclusions and Contributions

Conclusions

Contributions

Future Work

