Procedural Content Generation

Emmanuel Harley, Scott Spitzer, Preston Carroll

Background

Project

Goal: Provide a computational way to represent and operate on human workflows.

Purpose: To use virtual simulations to train AI in understanding workflow and provide technical fields a more effective training mechanism.

Our Team

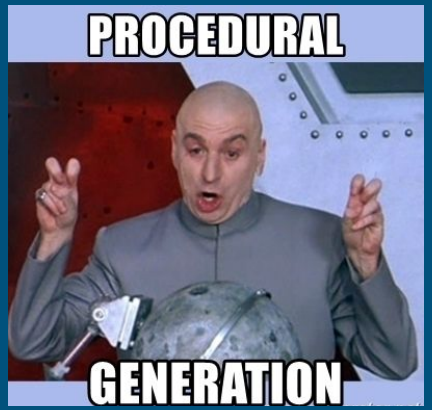
Goal: Procedurally generate viable environments where workflows are performed.

Purpose: Streamline the process of creating environments in a cheap and efficient manner

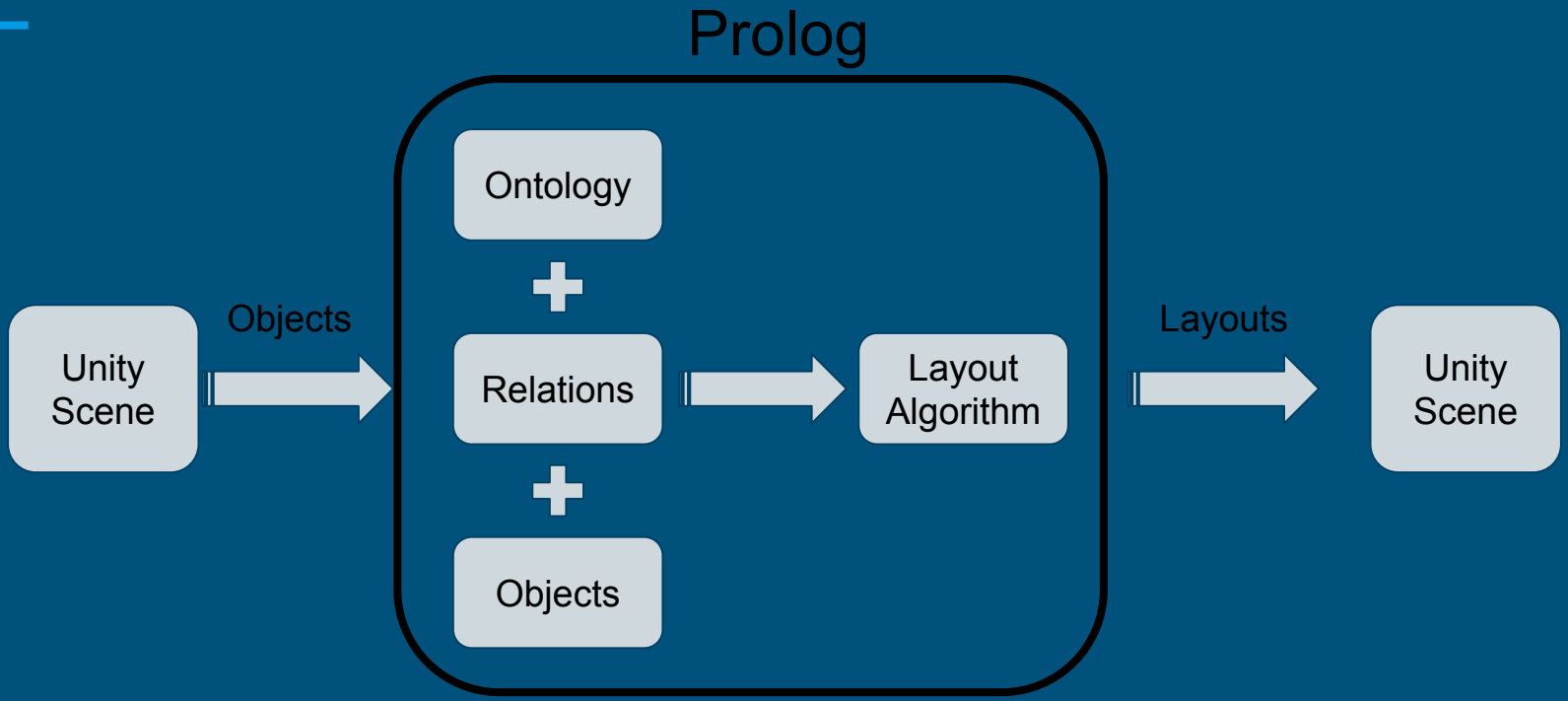
PCG Background

Methods:

- Cellular Automata
- Fractals
- Grammars and L-systems



Overview


Prolog
Ontology
Unity Scene
Objects
Relations
Layout Algorithm
Layouts
Unity Scene
Objects

Prolog - Answer Set Programming

Declarative programming paradigm used to determine a state that satisfies a problem statement

Facts - Truth Statements

Rules - Conditionals

State vs. Query

```
1 loves(romeo, rosaline).
2 loves(romeo, juliet).
3 loves(paris, juliet).
4 loves(juliet, romeo).
5
6 who_does_romeo_love(R) :-loves(romeo, R).
7 who_loves_romeo(R) :- loves(R, romeo).
8 who_loves_juliet(R) :- loves(R, juliet).
9 loves_each_other(R,Q) :- loves(R,Q), loves(Q,R).
```

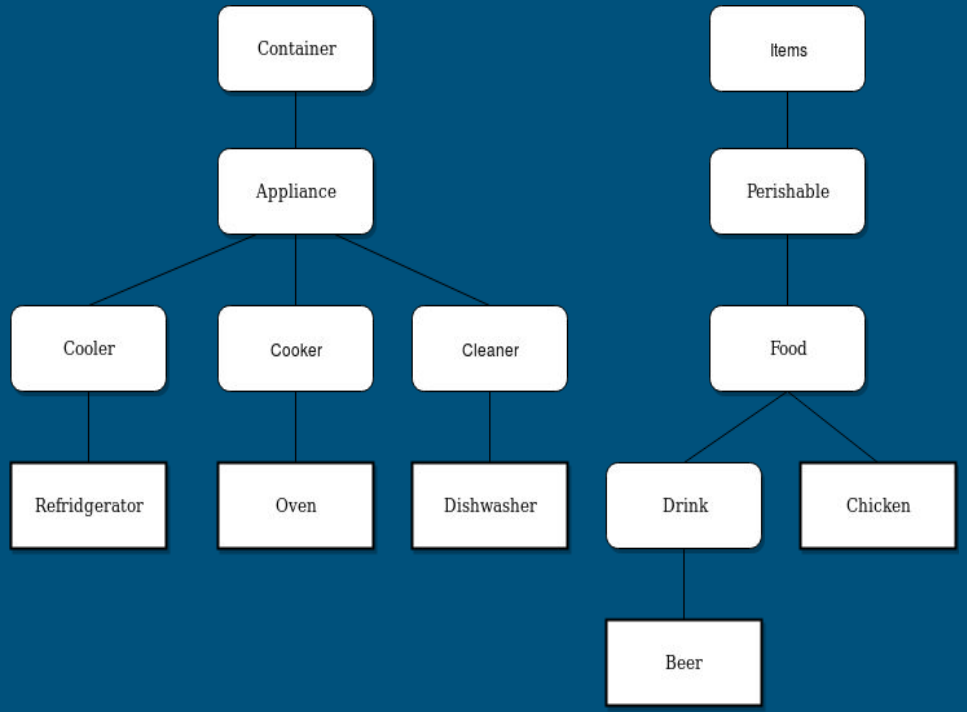
```
?- loves(rosaline, romeo).
false.

?- loves(romeo, juliet).
true.

?- loves(R, juliet).
R = romeo ;
R = paris.

?- who_does_romeo_love(R).
R = rosaline ;
R = juliet.

?- loves_each_other(M,F).
M = romeo,
F = juliet ;
M = juliet,
F = romeo ;
false.
```

Ontology

Entity relationship definitions

Effective in answer set programming

OWL: Specific language used to define ontologies



Quarterly Update

- Asset Pre-processing
- C# Prolog → SWI Prolog
- Absolute Position Model → Relational Model
- Expanded Ontology
  - More Objects
  - Greater hierarchy
  - More precise definitions

Asset Pre-Processing

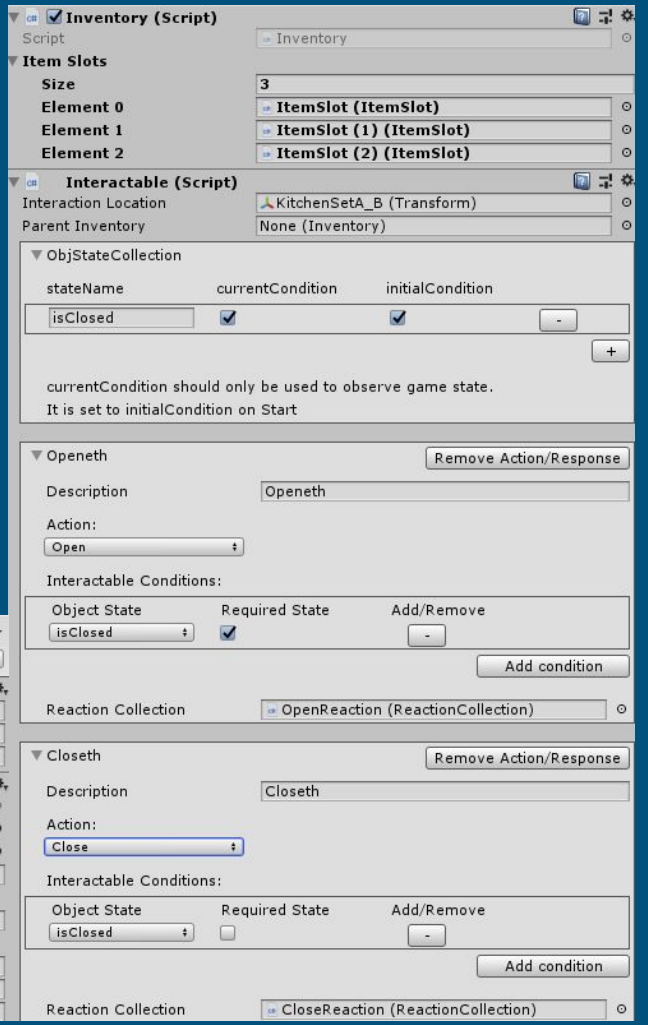
Definitions

Possible Actions

Animations

Reactions

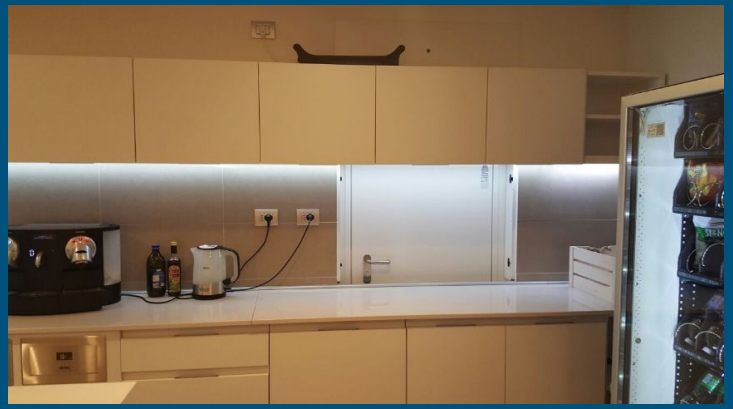
Locations



Relational Model

- Set of Rules
  - Feasibility
    - Can the object actually be placed there?
  - Existence
    - Can the relation exist without contradiction?
- Relations
  - “Left of”
  - “Behind”
  - “On top of”

Unity Scene Population



- Object Requirements

  Unity defines which objects should be in the room.
- Prolog Generation

  Generates set of valid layouts containing required objects.
- Unity Population

  Selects a random layout generated from Prolog and populates the scene.

Demo

Challenges

Massive change in approach to layout design

Unity - Placing objects based on a set of relations

Unity - Reworking models’ position data (sideways oven)

Meeting Times

Future Work

- More expansive and realistic relations
- Integrate OWL
- Semantic Image Segmentation via Ontology
- Simplify Preprocessing Assets
- Ontology editor
- Broaden range of environments
    - More rooms, hospitals

Tools

SWI/GNU Prolog Editor

Visual Studio (C#)

Unity

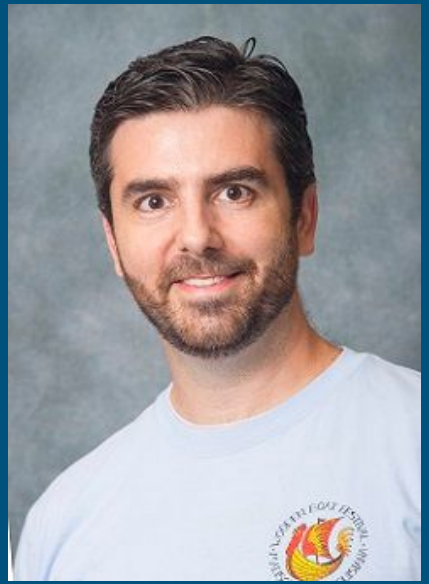
Draw.io

License

Acknowledgement

- Project Advisor & Client
  - Dr. Deneke

- Prolog Specialist
  - Dr. Bover



Questions