



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

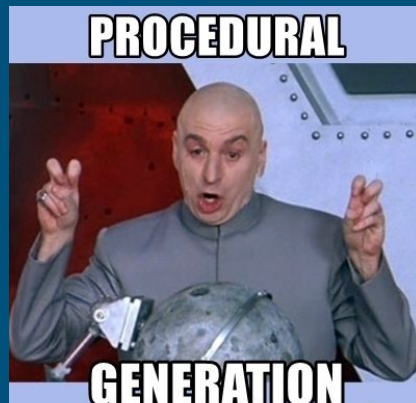
Goal: Procedurally generate viable environments where workflows are performed.

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

PCG Background

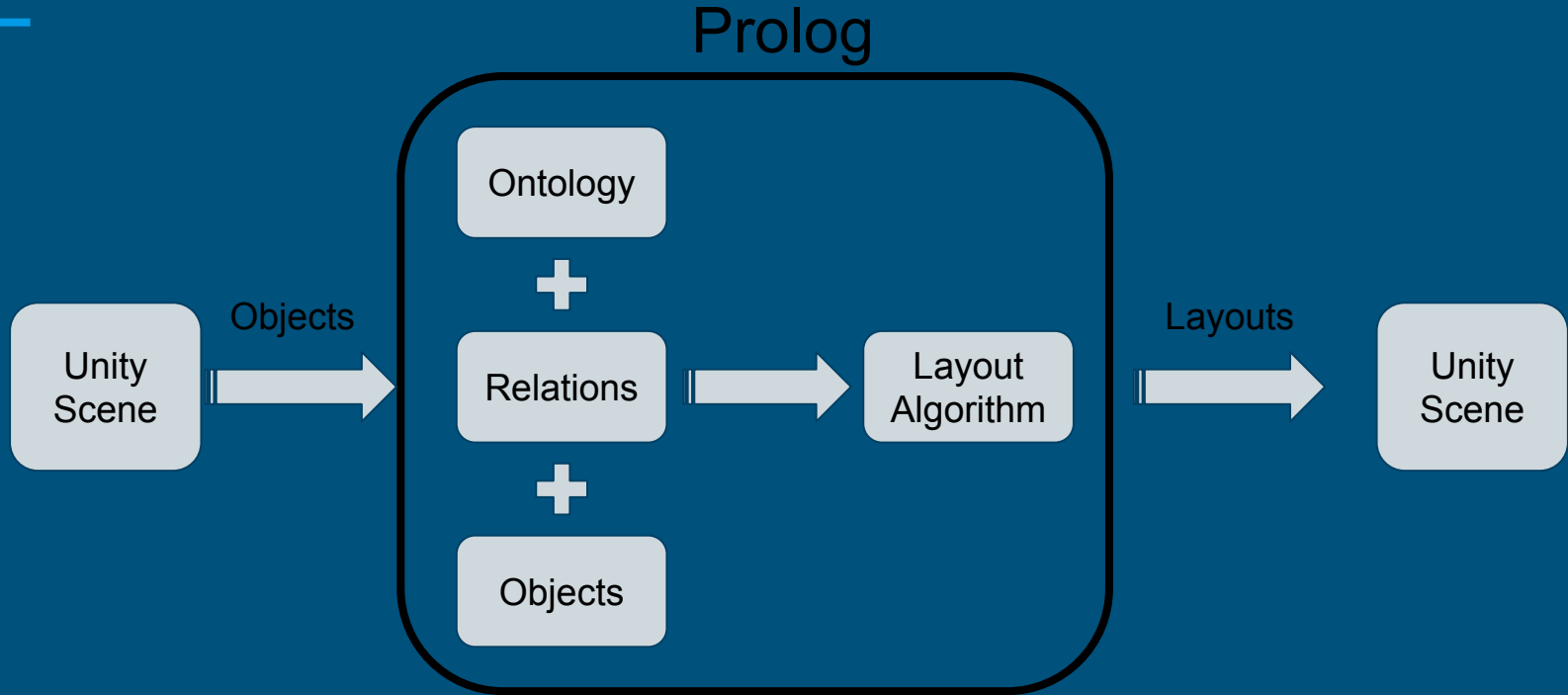
Methods:

- Cellular Automata
- Fractals
- Grammars and L-systems



LEVEL REQUIREMENT: 48 PRO CAUSTIC PROTECTOR Damage: 108 Accuracy: 72.8 Fire Rate: 4.3 Highly effective vs Armor Chance to Corrode enemies Removes no longer on issue +50% Reload Speed #12 x2 0103209	LEVEL REQUIREMENT: 48 BURG NASTY REBEL Damage: 137 Accuracy: 78.0 Fire Rate: 5.5 Never stop shooting! +39% Fire Rate +24 Magazine Size +11% Damage #36 x4 0099239	LEVEL REQUIREMENT: 48 GGN VOLCANO Damage: 439 Accuracy: 96.7 Fire Rate: 0.8 Highly effective vs Flesh Chance to light enemies on fire Pole demands a sacrifice High Elemental Effect chance #46 x4 0165399	LEVEL REQUIREMENT: 48 RPG12 NIDHOGG Damage: 1771 Accuracy: 95.1 Fire Rate: 0.7 Death rains from above +52% Recall Reduction +41% Accuracy Rebust Rockets #12 HYPERION
LEVEL REQUIREMENT: 48 SG30 BLAST HAMMER Damage: 15 x7 Accuracy: 23.0 Fire Rate: 1.8 Chance to cause explosions Drop the Hammer! +31% Fire Rate #7 x2	LEVEL REQUIREMENT: 48 GGN VOLCANO Damage: 439 Accuracy: 96.7 Fire Rate: 0.8 Highly effective vs Flesh Chance to light enemies on fire Pole demands a sacrifice High Elemental Effect chance #46 x4	LEVEL REQUIREMENT: 48 DEF11 TERRIBLE DEFENDER Damage: 51.3 Accuracy: 51.3 Fire Rate: 1.0 I can do this all day... +18 Ammo Regeneration +59% Reload Speed -2% Damage #6 x4	LEVEL REQUIREMENT: 48 CR380 DESERT RAVEN Damage: 183 Accuracy: 91.7 Fire Rate: 10.0 2 more bullets make all the difference +22% Recall Reduction 1.4x Weapon Zoom #21 DAHL
LEVEL REQUIREMENT: 48 ED2 RAW EQUALIZER Damage: 371 Accuracy: 63.9 Fire Rate: 1.9 Unending Firepower +18 Ammo Regeneration +57% Reload Speed +33% Fire Rate #6	LEVEL REQUIREMENT: 48 KLR13 PESTILENT DEFILER Damage: 541 Accuracy: 93.6 Fire Rate: 1.0 Highly effective vs Armor Chance to Corrode enemies Give Sick High Elemental Effect chance #3 x4	LEVEL REQUIREMENT: 48 ED1 STABILIZED EQUALIZER Damage: 383 Accuracy: 95.1 Fire Rate: 1.5 Unending Firepower +41% Reload Speed +18 Ammo Regeneration +50% Recall Reduction #6	LEVEL REQUIREMENT: 48 TD2 DOUBLE BITCH Damage: 100 Accuracy: 50.3 Fire Rate: 8.3 Smack 'em 3.9x Weapon Zoom +55% Recall Reduction +42% Accuracy #33 HYPERION
LEVEL REQUIREMENT: 48 CR280 DESERT RAVEN Damage: 213 Accuracy: 90.4 Fire Rate: 10.0 2 more bullets make all the difference 2.4x Weapon Zoom +25% Recall Reduction #21 DAHL	LEVEL REQUIREMENT: 48 SG330 BLAST BUTCHER Damage: 7 x3 Accuracy: 58.8 Fire Rate: 8.8 Chance to cause explosions Rhh... Fresh meat! +500% Burst Fire Count +41% Recall Reduction #7 x2	LEVEL REQUIREMENT: 48 SV440 MALEVOLENT SMURF Damage: 134 Accuracy: 95.0 Fire Rate: 14.6 Hole-in-one +18 Ammo Regeneration +19% Reload Speed +23% Damage #55 TEDIOR	LEVEL REQUIREMENT: 48 MAL31 PESTILENT DEFILER Damage: 371 Accuracy: 96.4 Fire Rate: 1.6 Highly effective vs Armor Chance to Corrode enemies Give Sick 3.7x Weapon Zoom #6 x4
0381170	0436059	0446578	0390756

Overview



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 ;
```

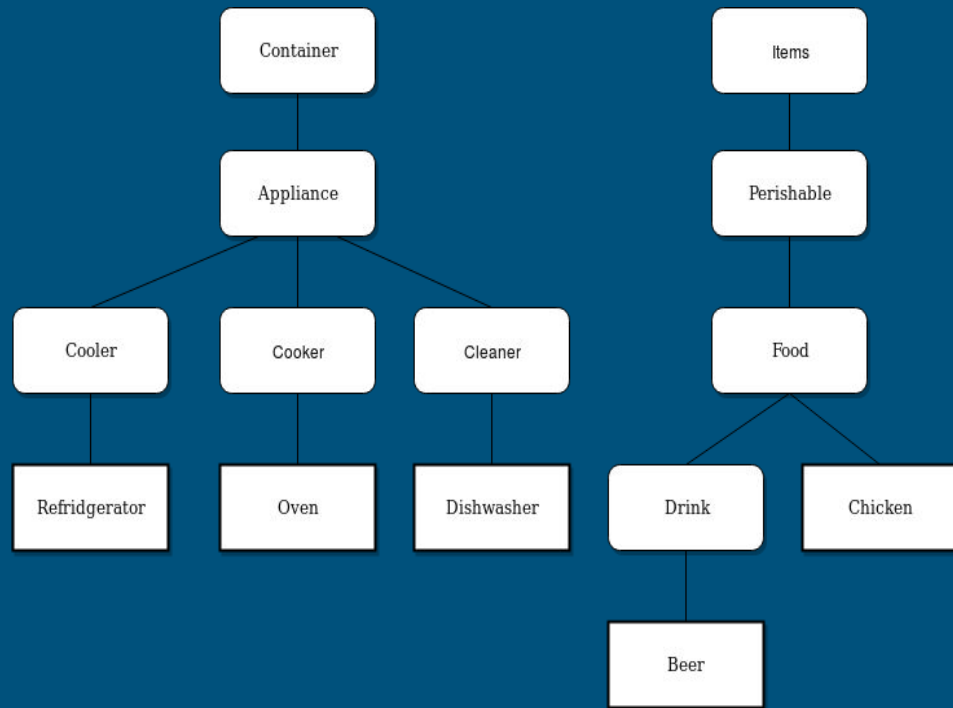
```
false.
```

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

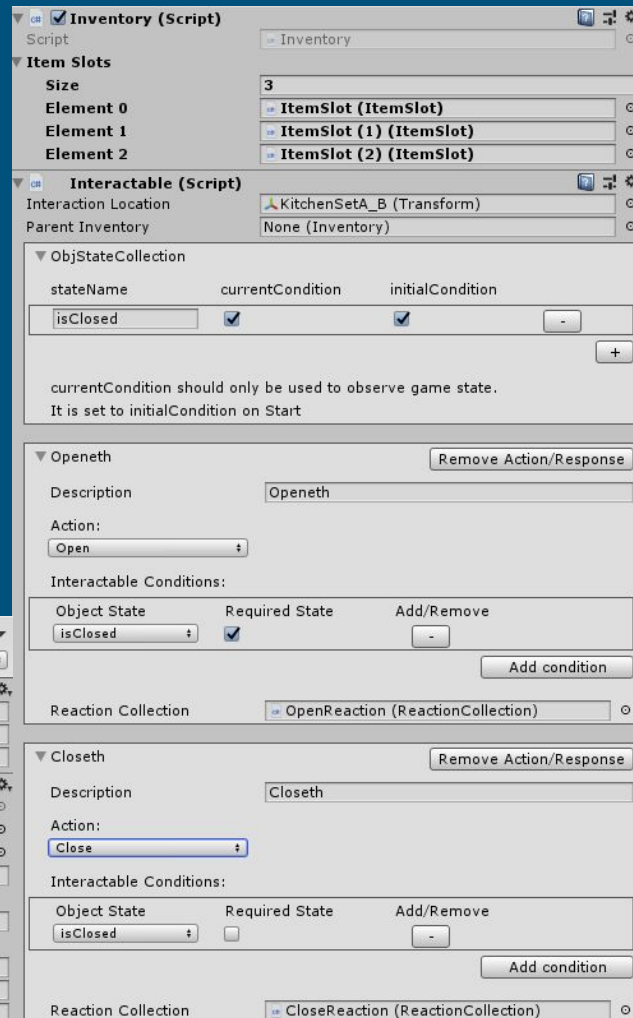
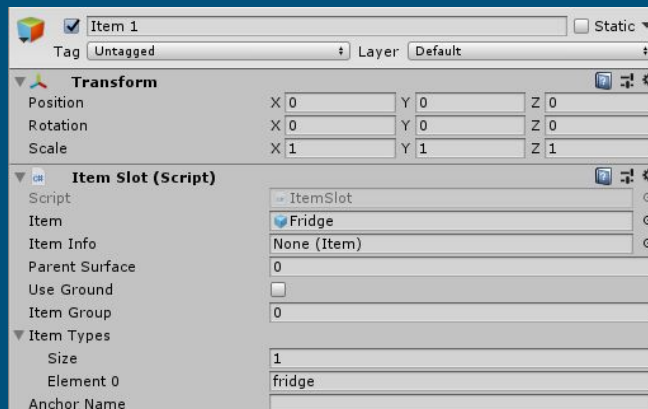
Definitions

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

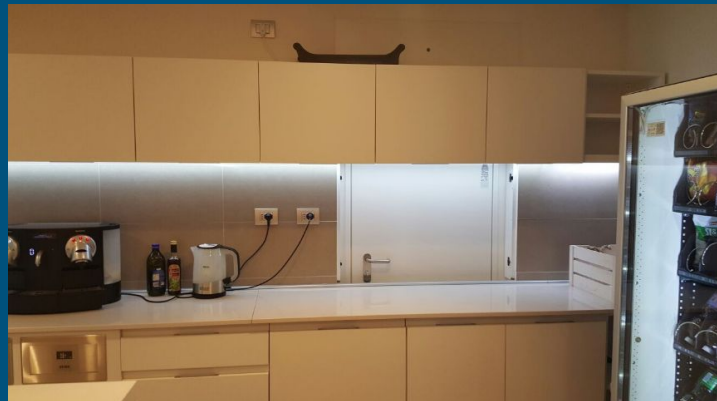
Unity defines which objects should be in the room.

- 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

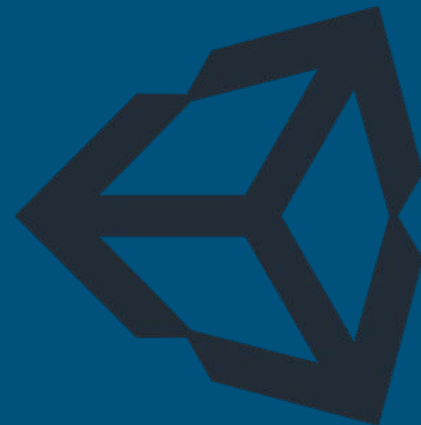
Draw.io



SWI Prolog



draw.io



License



License

Acknowledgement

- Project Advisor & Client
 - Dr. Deneke
- Prolog Specialist
 - Dr. Bover



Questions
