Centrum Wiskunde & Informatica

Hogeschool van Amsterdam
Amsterdam University of Applied Sciences

Live Game Design

RAAK-MKB project

CWI Scientific Meeting – March 31st 2017

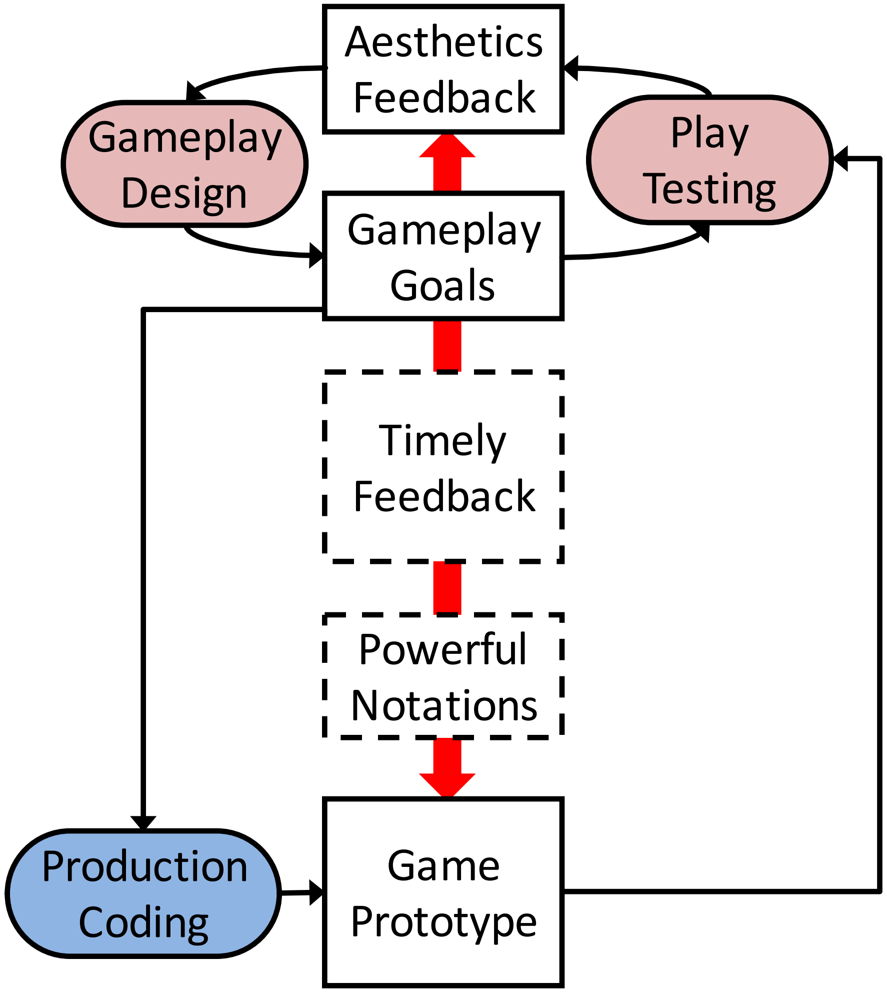
Riemer van Rozen
CWI SWAT group

Joint work with:

Tijs van der Storm, Paul Klint

Problem Statement

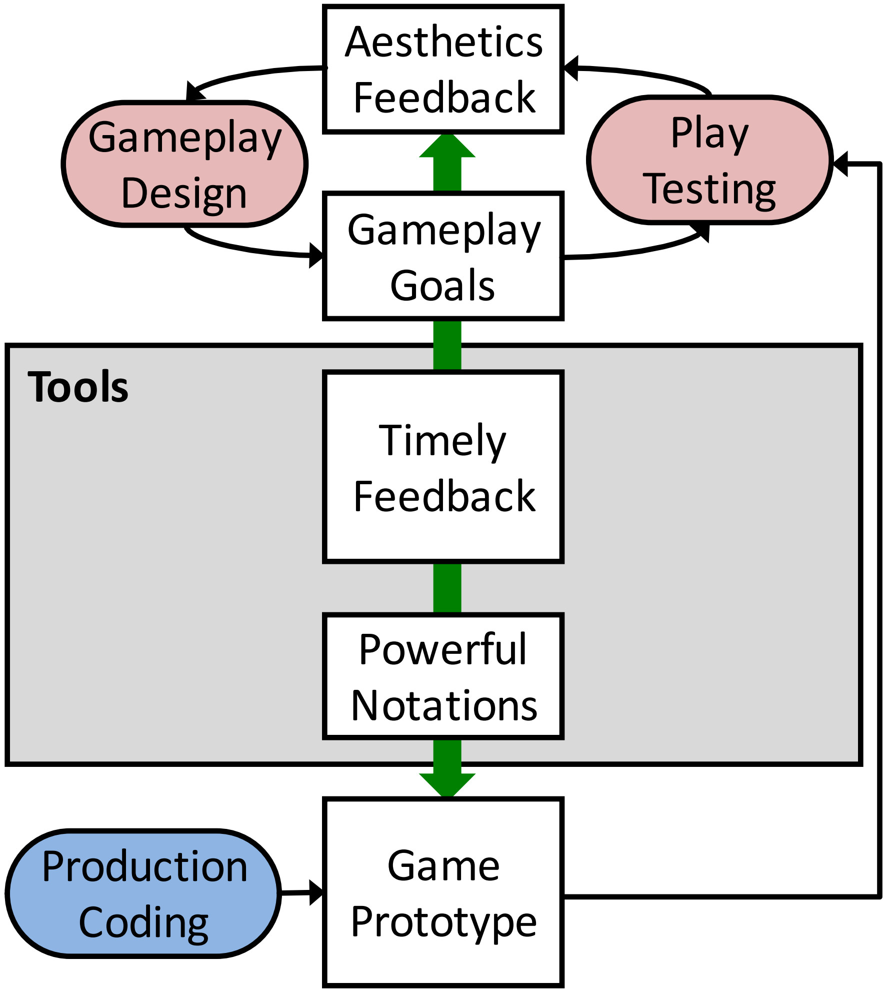
- **Problem**
  - representation gap of game design: ***"the gap between a game's design and its source code"***
- **Long game design iterations**
  - prevent quickly experimenting with alternative game designs
  - game quality under pressure
- **Missing**
  - powerful notations for modifying a game's elements
  - timely feedback



Objectives

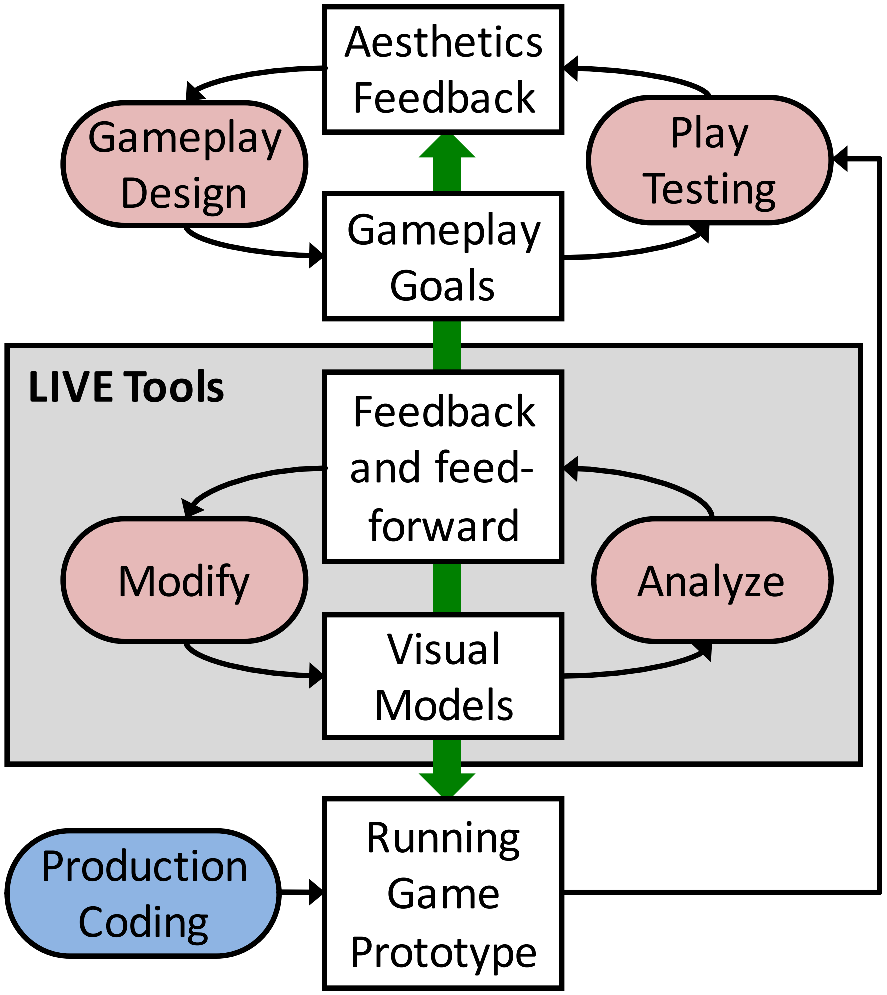
- **Question**
  - Can the representation gap of game design be bridged with tools for exploring the design space?

- **How can tools help**
  1. shorten game design iterations and speed-up the design process
  2. closely match design to expertise and imagination
  3. help to improve the quality
  4. enable to design in a more targeted way



Approach: Live Game Design

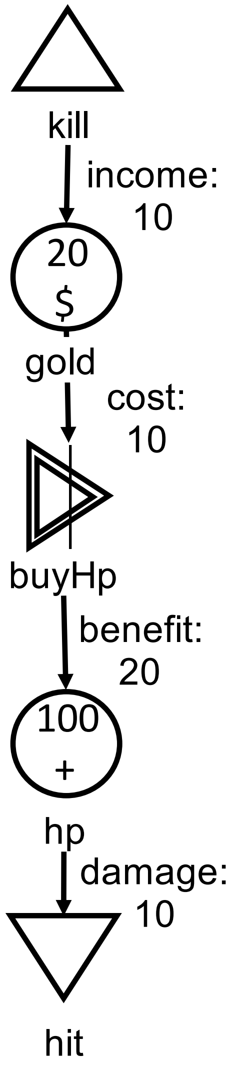
- **Approach**
  - Live Intelligent Visual Environments for Game Design (Live Game Design)
- **Visual Programming Languages**
  - Visual notations for describing and steering interactive game elements (prototyping, fine-tuning) attuned to the expertise of game designers
- **Live feedback and feed-forward**
  - Immediate and continuous feedback on modification results
  - Design alternatives that can be inspected and applied to focus the creative design process



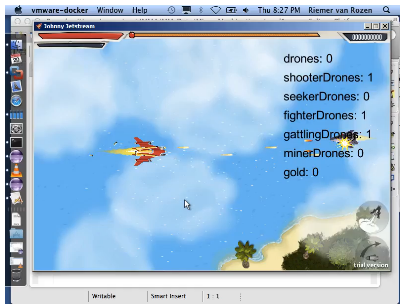
Live Textual Domain-Specific Languages

- **Domain-Specific Language (DSL) for the Game Domain:** Micro-Machinations is a language and library that enables game designers to modify a game's rules at run-time.
- **Example:** Johnny Jetstream

```
source kill
income: kill -10-> gold
pool gold is "$" at 20
cost: gold -10-> buyHp
user converter buyHp
benefit: buyHp -20-> hp
pool hp is "+" at 100
damage: hp -10-> hit
drain hit
```

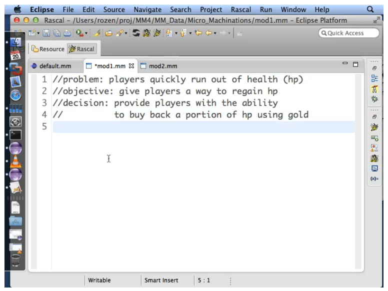


Step 1: Play Test v1



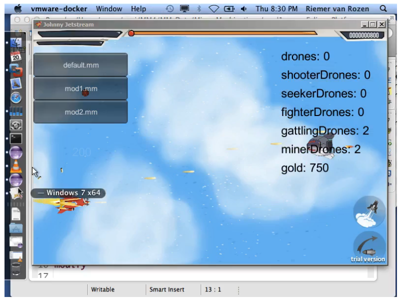
```
drones: 0
shooterDrones: 1
seekerDrones: 0
fighterDrones: 1
gattlingDrones: 1
minerDrones: 0
gold: 0
```

Step2: Re-design



```
//problem: players quickly run out of health (hp)
//objective: give players a way to regain hp
//decision: provide players with the ability
//          to buy back a portion of hp using gold
```

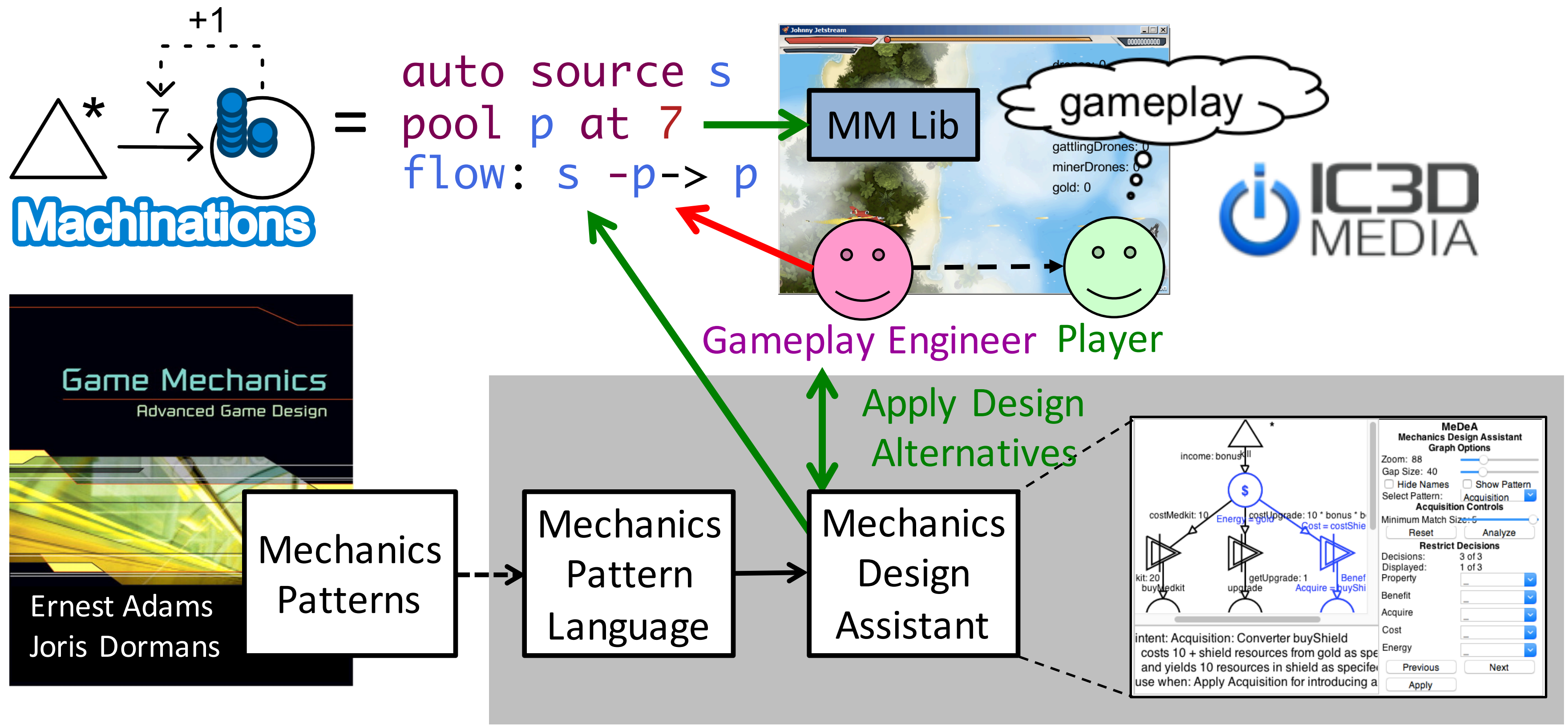
Step 3: Play Test v2



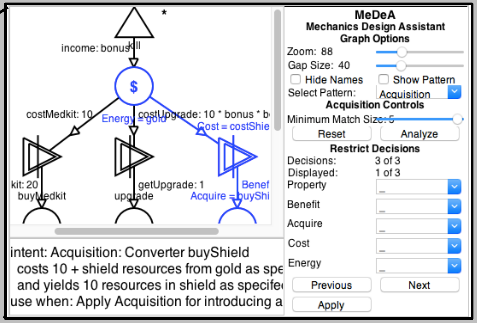
```
drones: 0
shooterDrones: 0
seekerDrones: 0
fighterDrones: 0
gattlingDrones: 2
minerDrones: 2
gold: 750
```

Machinations Evolution & Approach

2009 Conceptual Game Design Aid → **2013** Formal Analysis + text, modules → **2014** Live Adaptations → **2015** A Pattern-Based Game Mechanics Design Assistant



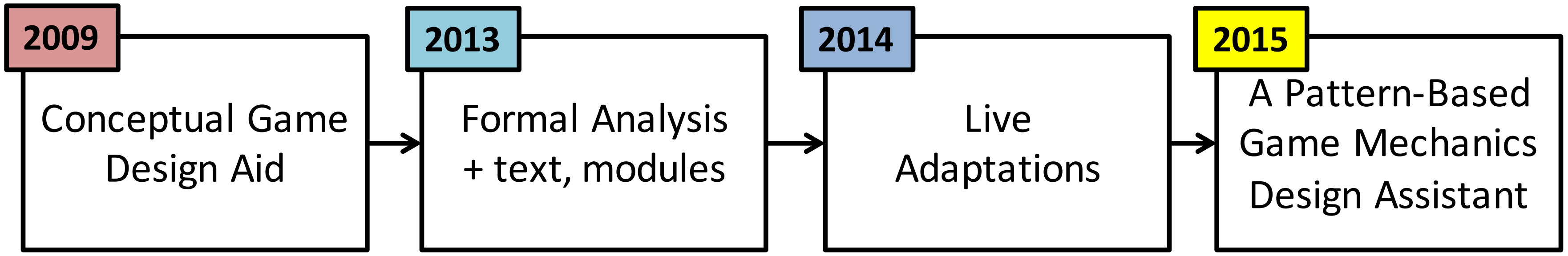
```
auto source s
pool p at 7
flow: s -p-> p
```



Zoom: 88
Gap Size: 40
Select Pattern: Acquisition
Minimum Match Size: 5
Decisions: 3 of 3
Displayed: 1 of 3
Property
Benefit
Acquire
Cost
Energy

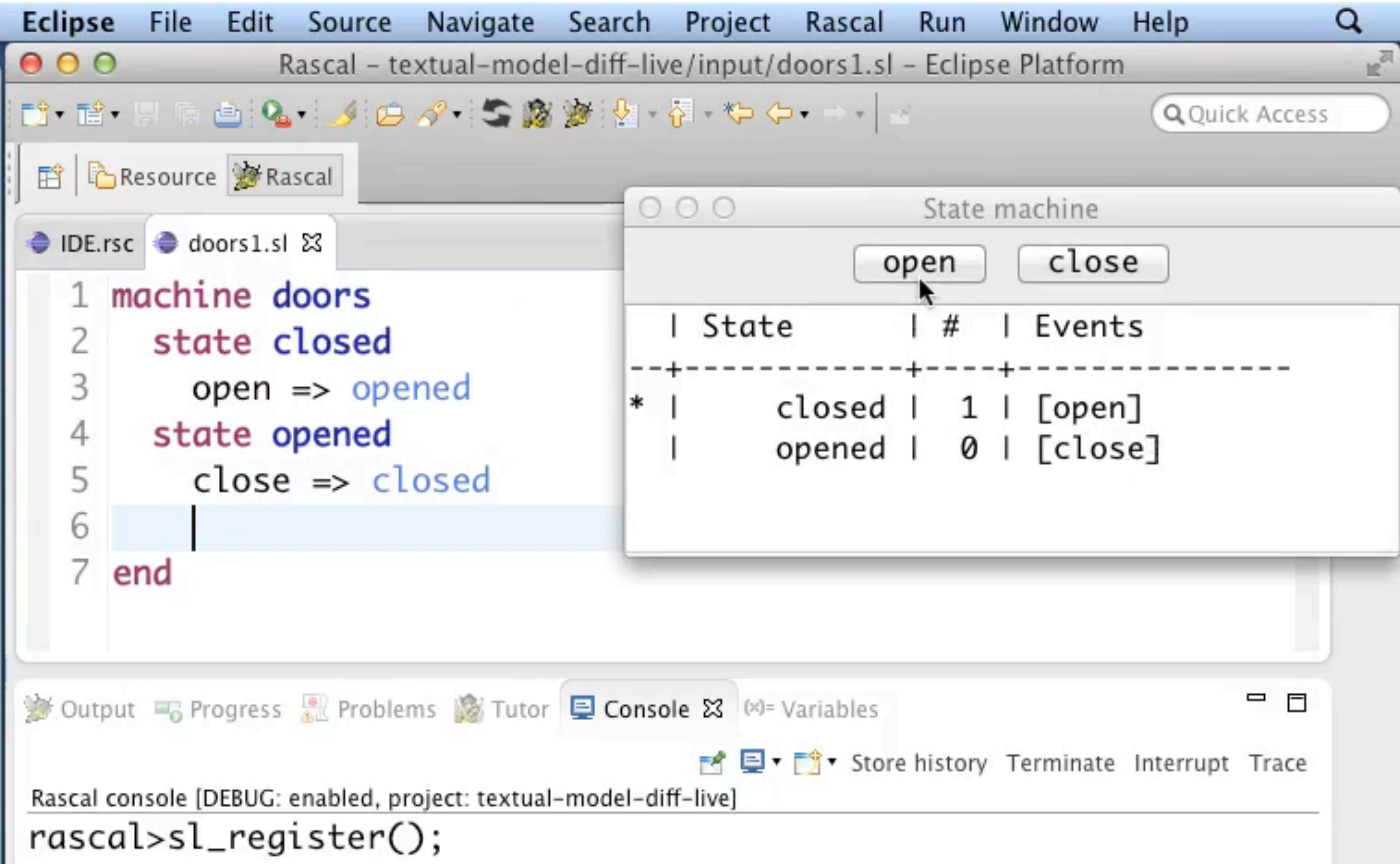
intent: Acquisition: Converter buyShield
costs 10 + shield resources from gold as spe
and yields 10 resources in shield as specife
use when: Apply Acquisition for introducing a

Machinations Evolution & Approach


2009
Conceptual Game Design Aid
2013
Formal Analysis + text, modules
2014
Live Adaptations
2015
A Pattern-Based Game Mechanics Design Assistant

- J. Dormans. Machinations: Elemental Feedback Patterns for Game Design, In International North American Conference on Intelligent Games and Simulation, 2009.
- E. Adams and J. Dormans. Game Mechanics: Advanced Game Design. New Riders Publishing, Thousand Oaks, CA, USA, 1st edition, 2012.
- P. Klint and R. van Rozen. Micro-Machinations: A DSL for Game Economies. In Software Language Engineering, 2013.
- R. van Rozen and J. Dormans. Adapting Game Mechanics with Micro-Machinations. In Foundations of Digital Games, 2014.
- R. van Rozen. A Pattern-Based Game Mechanics Design Assistant. In Foundations of Digital Games, 2015.

Live State Machine Language in Rascal



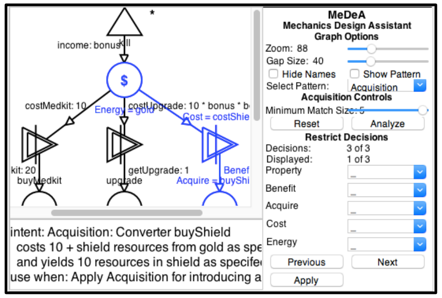
```
machine doors
  state closed
    open => opened
  state opened
    close => closed

end
```

```
  | State       | #  | Events
--+-------------+----+---------------
* |      closed |  1 | [open]
  |      opened |  0 | [close]
```

```
Rascal console [DEBUG: enabled, project: textual-model-diff-live]
rascal>sl_register();
```

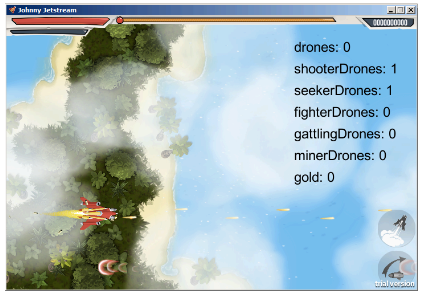
Current and Future Work

- **Question**
  - Can the representation gap of game design be bridged with tools for exploring the design space?
- **Approach**
  - Live Intelligent Visual Environments for Game Design (Live Game Design)
- **Work in progress**
  - Live Game Design project http://livegamedesign.github.io/
  - Generic frameworks for Live DSLs
- **Liked the live programming demo?**
  - Twitter @rvrozen



Zoom: 88
Gap Size: 40
Select Pattern: Acquisition
Minimum Match Size: 5
Decisions: 3 of 3
Displayed: 1 of 3

intent: Acquisition: Converter buyShield
costs 10 + shield resources from gold as spe
and yields 10 resources in shield as specife
use when: Apply Acquisition for introducing a



drones: 0
shooterDrones: 1
seekerDrones: 1
fighterDrones: 0
gattlingDrones: 0
minerDrones: 0
gold: 0

References

- T. van der Storm. Semantic Deltas for Live DSL Environments. In workshop on Live Programming, 2013.
- R. van Rozen and J. Dormans. Adapting Game Mechanics with Micro-Machinations. In Foundations of Digital Games, 2014.
- R. van Rozen, T. van der Storm. Origin Tracking + Text Differencing = Textual Model Differencing. In Theory and Practice of Model Transformations 2015.