Automated Game Design

Michael Cook
Queen Mary University of London

@mtrc
possibilityspace.org
mike@gamesbyangelina.org

Hi!

- Building AI that do **creative** things, like designing games.
- Developing tools and techniques to **understand generative systems**
- Finding new applications for procedural generation, especially **new kind of game experience.**

Contact me any time!

@mtrc on Twitter
mike@gamesbyangelina.org

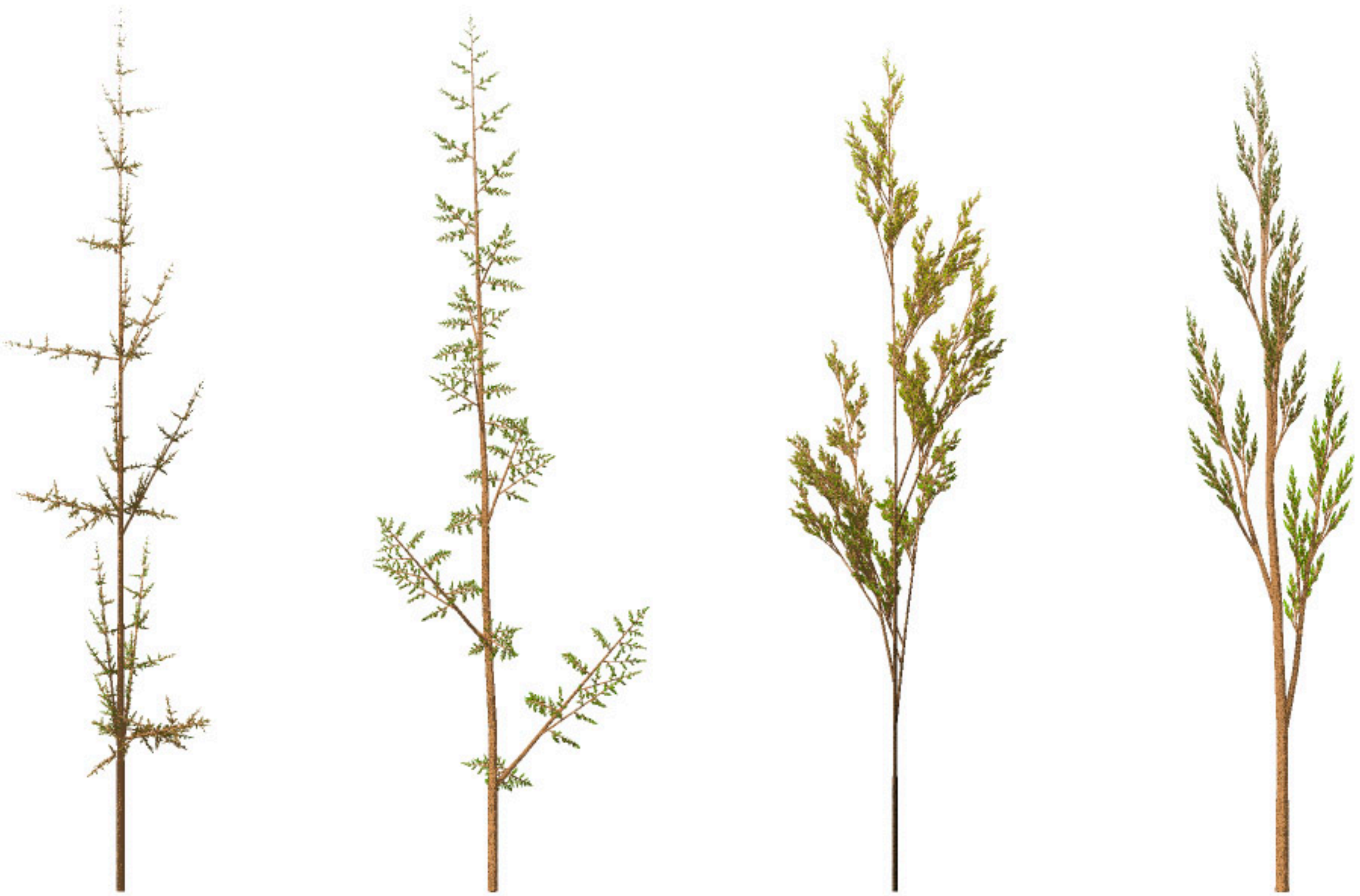
Automated Game Design

- A non-exhaustive history of Automated Game Design
- A grounding in how to analyse and criticise AGD systems
- A little insight into how philosophies shape research
- An idea of the future of the field and how you can join

Generative Systems

Tarot cards

Generative Systems

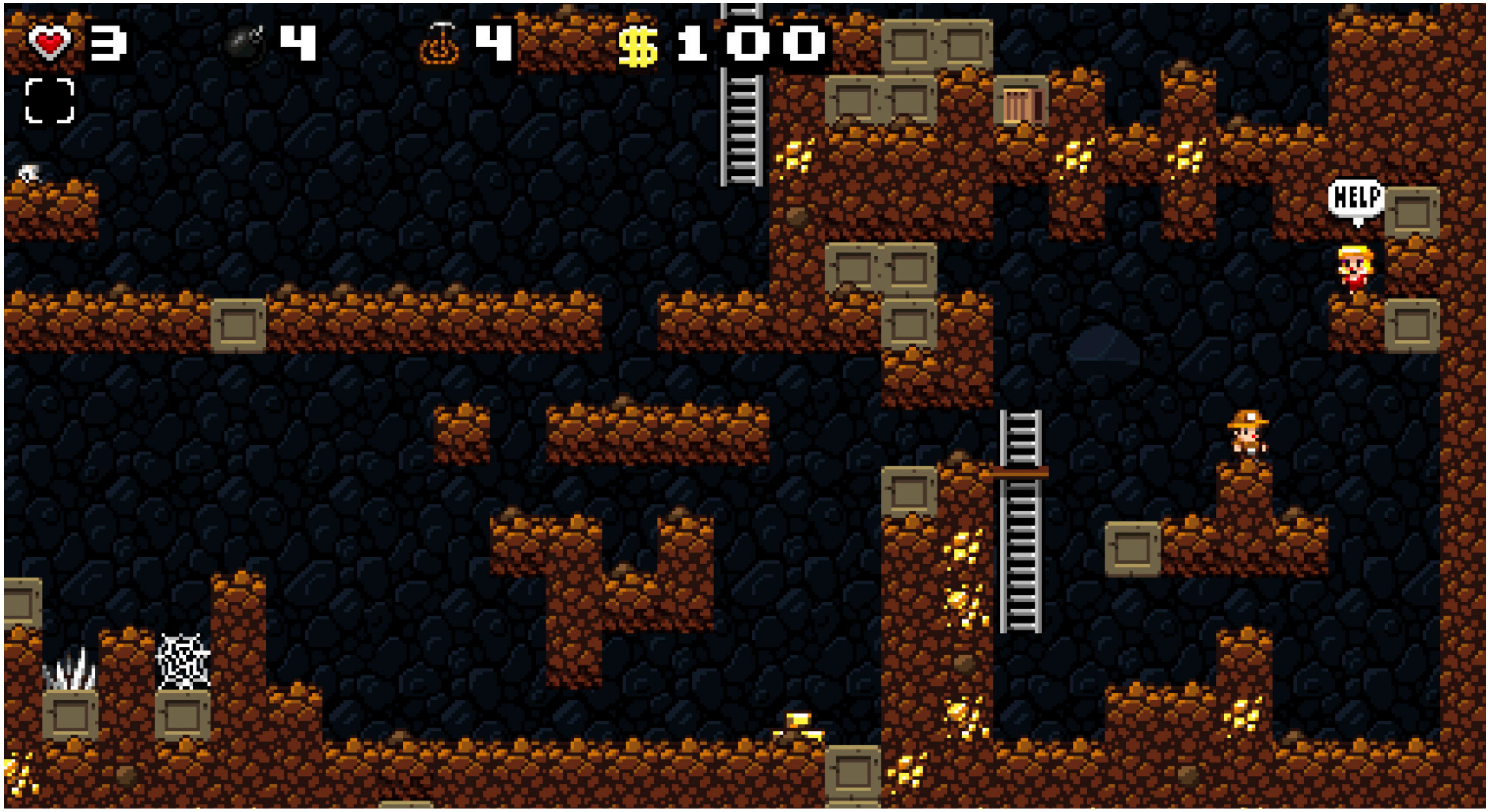


Trees made with L-Systems

Procedural Generation

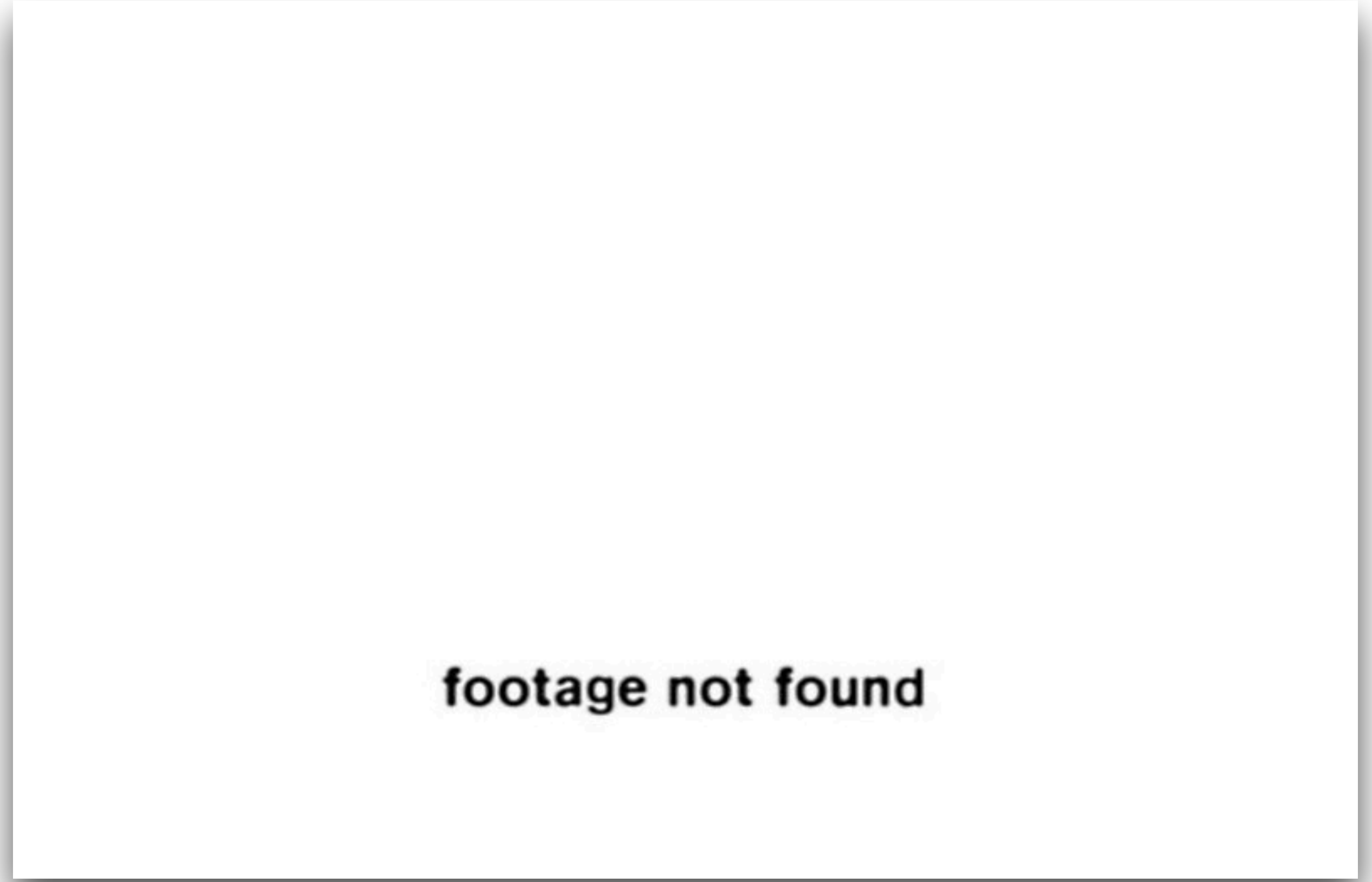
Elite (1984)

Procedural Generation



Spelunky (2008)

Automated Game Design



footage not found

???

1st Generation

1992-2010

2nd Generation

2010-2016

3rd Generation

2016-????

METAGAME

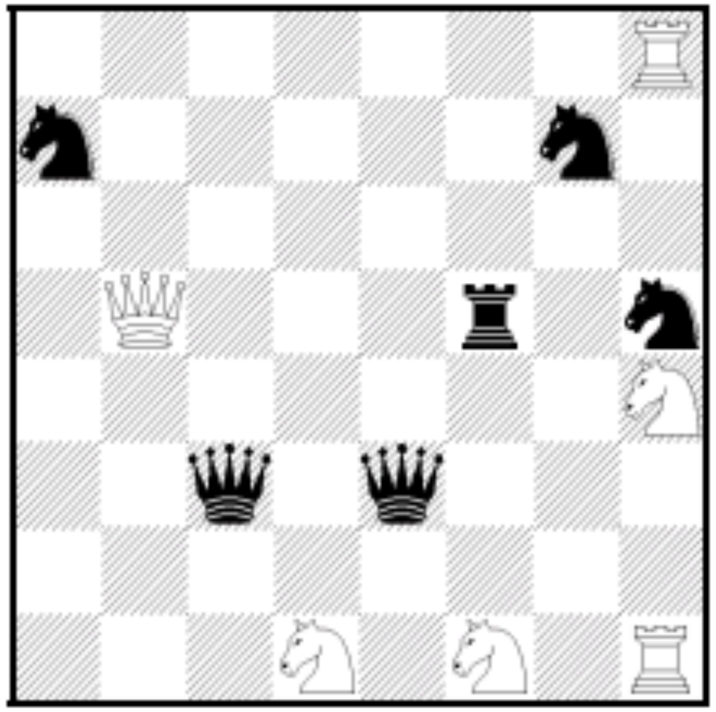
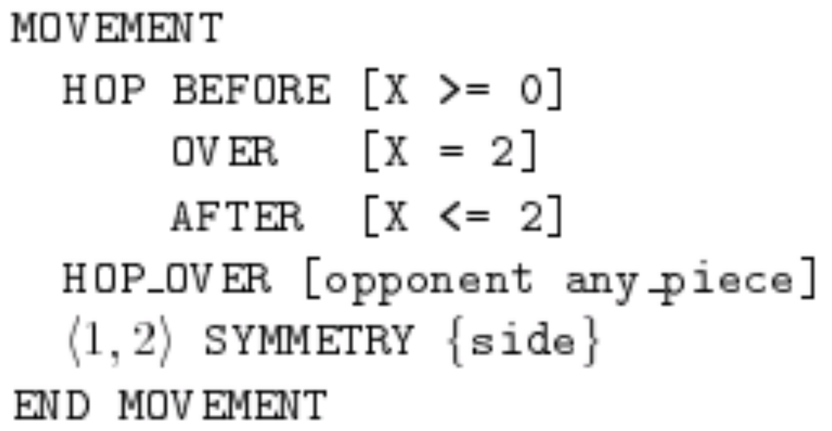


Figure 2: Example piece movements.



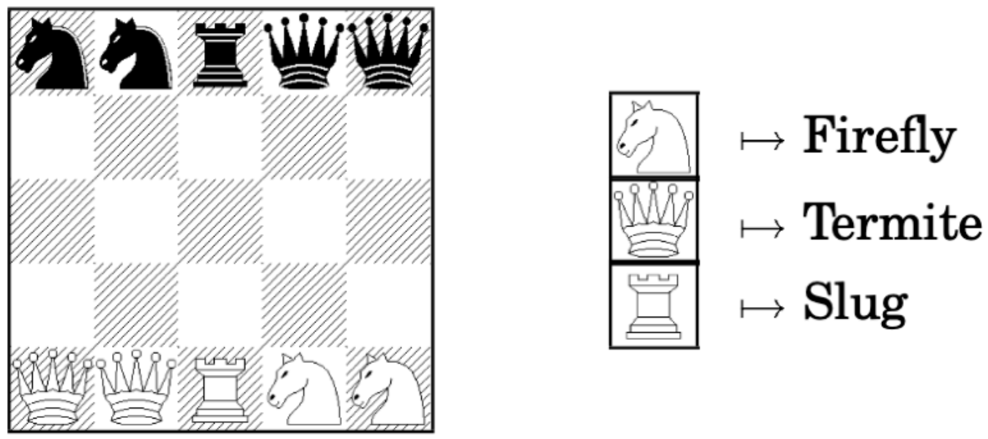
```
MOVEMENT
  HOP BEFORE [X >= 0]
      OVER   [X = 2]
      AFTER  [X <= 2]
  HOP_OVER [opponent any_piece]
  ⟨1,2⟩ SYMMETRY {side}
END MOVEMENT
```

Pell, B. *METAGAME in Symmetric Chess-Like Games* (1992)

Pell developed a system to invent variants on Chess, with an expressive design language.

Very limited filtering: “*the generator performs a simple analysis to avoid… a high proportion of trivial games*”

Example: *Turncoat Chess*

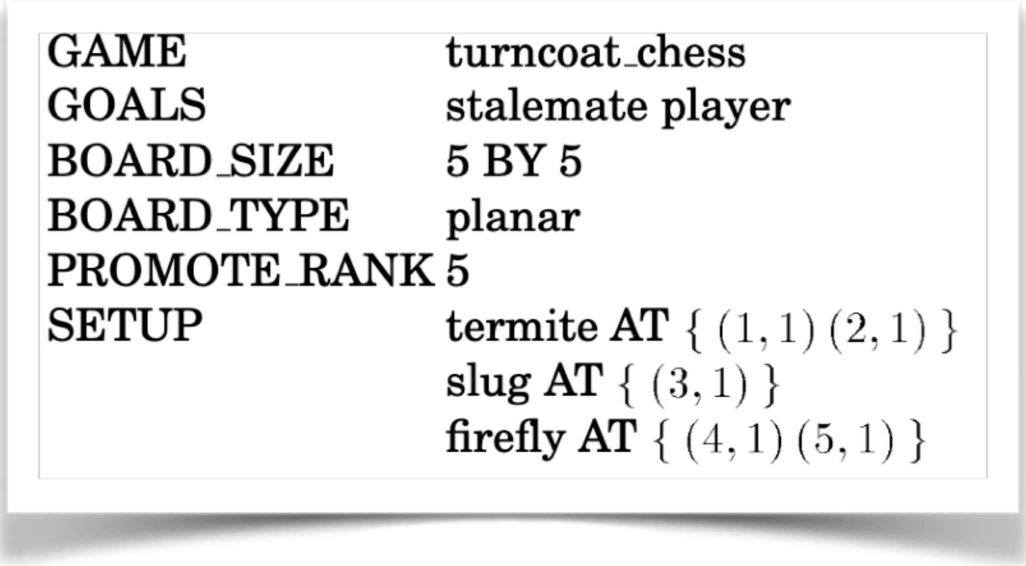


3 piece types, can capture both friendly and enemy pieces

Pieces can be ‘promoted’ into other types, and some of these automatically give the piece to the other player.

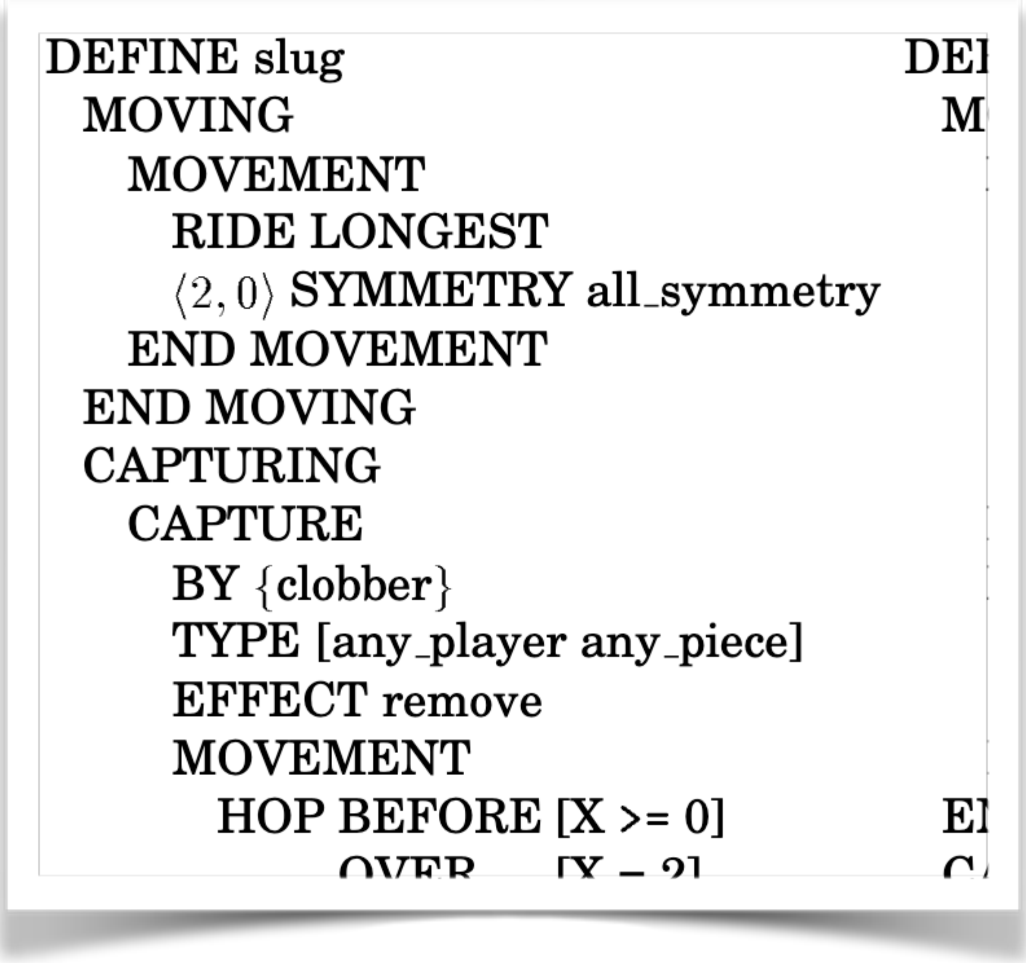
The first player to run out of legal moves wins.

Pell created a *game description language* which all of METAGAME's games are defined in.



```
GAME            turncoat_chess
GOALS           stalemate player
BOARD_SIZE      5 BY 5
BOARD_TYPE      planar
PROMOTE_RANK 5
SETUP           termite AT { (1, 1) (2, 1) }
                slug AT { (3, 1) }
                firefly AT { (4, 1) (5, 1) }
```

Note how the **structure** of a game is quite strictly defined, METAGAME mixes up the **details**.



```
DEFINE slug                                 DEI
  MOVING                                     M
    MOVEMENT
      RIDE LONGEST
      ⟨2, 0⟩ SYMMETRY all_symmetry
    END MOVEMENT
  END MOVING
  CAPTURING
    CAPTURE
      BY {clobber}
      TYPE [any_player any_piece]
      EFFECT remove
      MOVEMENT
        HOP BEFORE [X >= 0]                 EI
             OVER     [X = 2]               CA
```

The rules are almost exclusively contained within the descriptions of the pieces themselves.

Concepts: Design Space

The **design space** of an AGD system is the set of all games that the system can express or describe.

When we evaluate an AGD system, it’s useful to ask **what is and isn’t in the design space**.

Some design spaces are too large - e.g. generating control schemes

The size and shape of the design space governs the difficulty of the search problem.

Curation

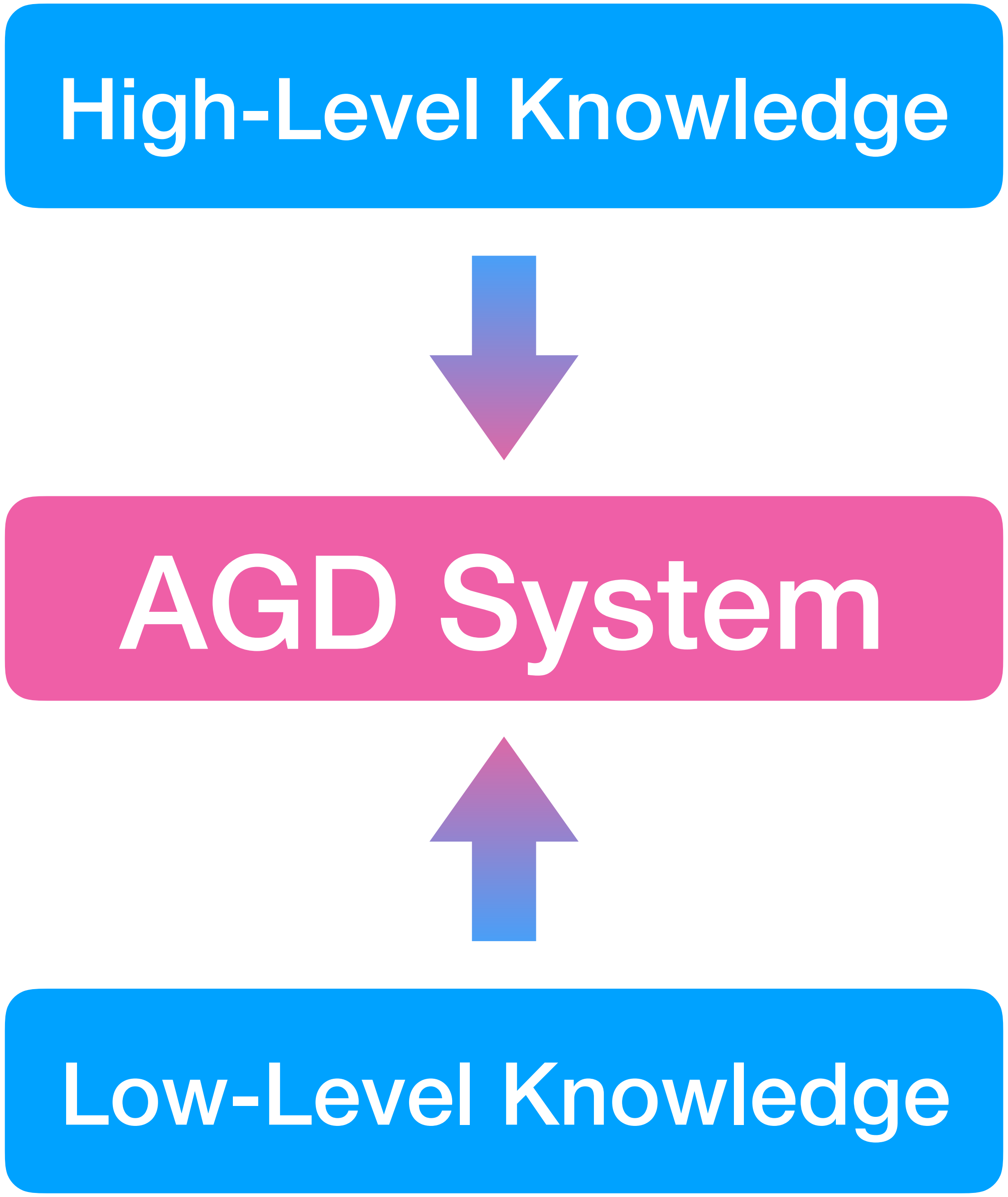
Question:
How rare are games like Turncoat Chess?

Most AGD papers include example games to showcase the system’s output, usually cherrypicked to be interesting or good.

We define the **curation coefficient** as the percentage of outputs from a system that the creator would be happy showing to others.

If every output from METAGAME is high-quality, that’s great!
If 0.1% of them are high-quality, that’s less great

Embedded Design Knowledge



High-level features control the general shape of games made by the system. e.g. *the concept of pieces*

Low-level features are specific, atomic design concepts that affect the granularity of the system. e.g. *stalemate or in-a-row*

Critically Evaluating AGD Systems

Design Space

What games can this system express?
How hard is it to find a good game in its design space?

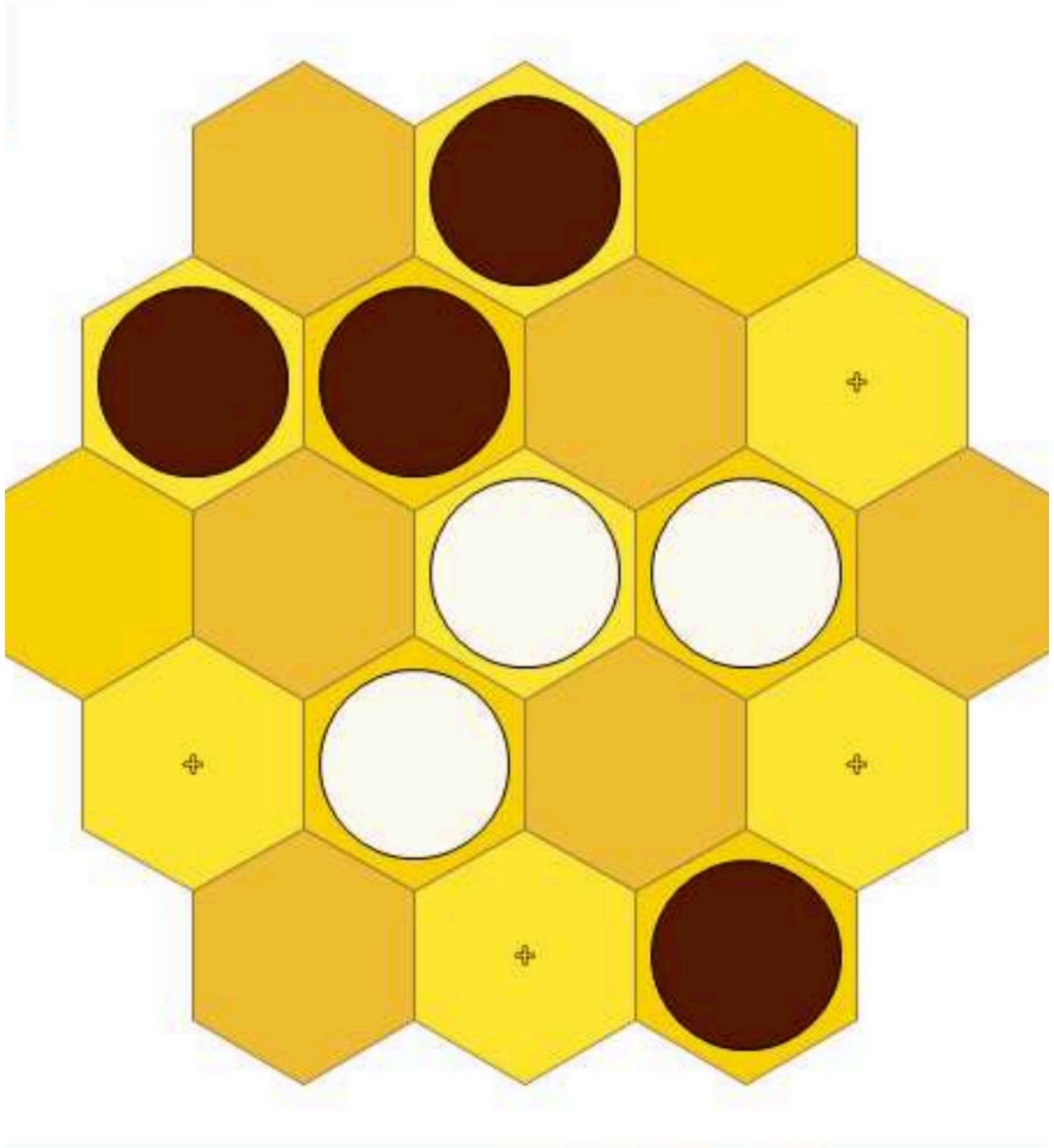
Curation

How much human curation is needed?
How have the authors shown this?

Embedded Knowledge

How much knowledge has been embedded by the designers?

Ludi



Browne, C and Maire, F.
Evolutionary Game Design (2010)

Browne and Maire describe Ludi, a system that invents and tests abstract boardgames.

Main step forward from METAGAME: Ludi can play games to test them. Browne and Maire define ways to measure what they see as ‘good’ qualities of a game design.

Example: Yavalath

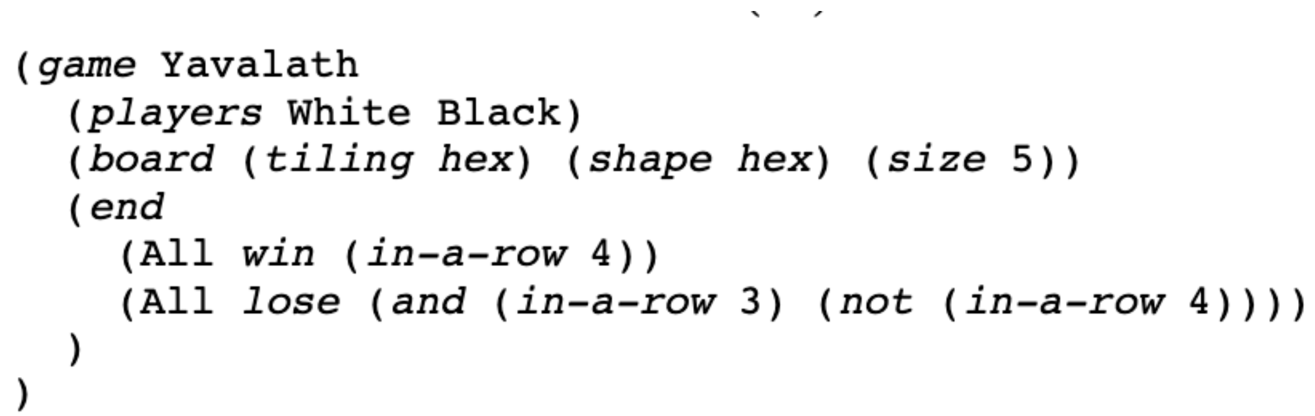
A 2-3 player hex-based game.

Players take turns to place pieces on the board. The first player to make a row of 4 wins. If a player makes a row of 3, they lose.

Players can be forced into making a losing move to block a winning one.

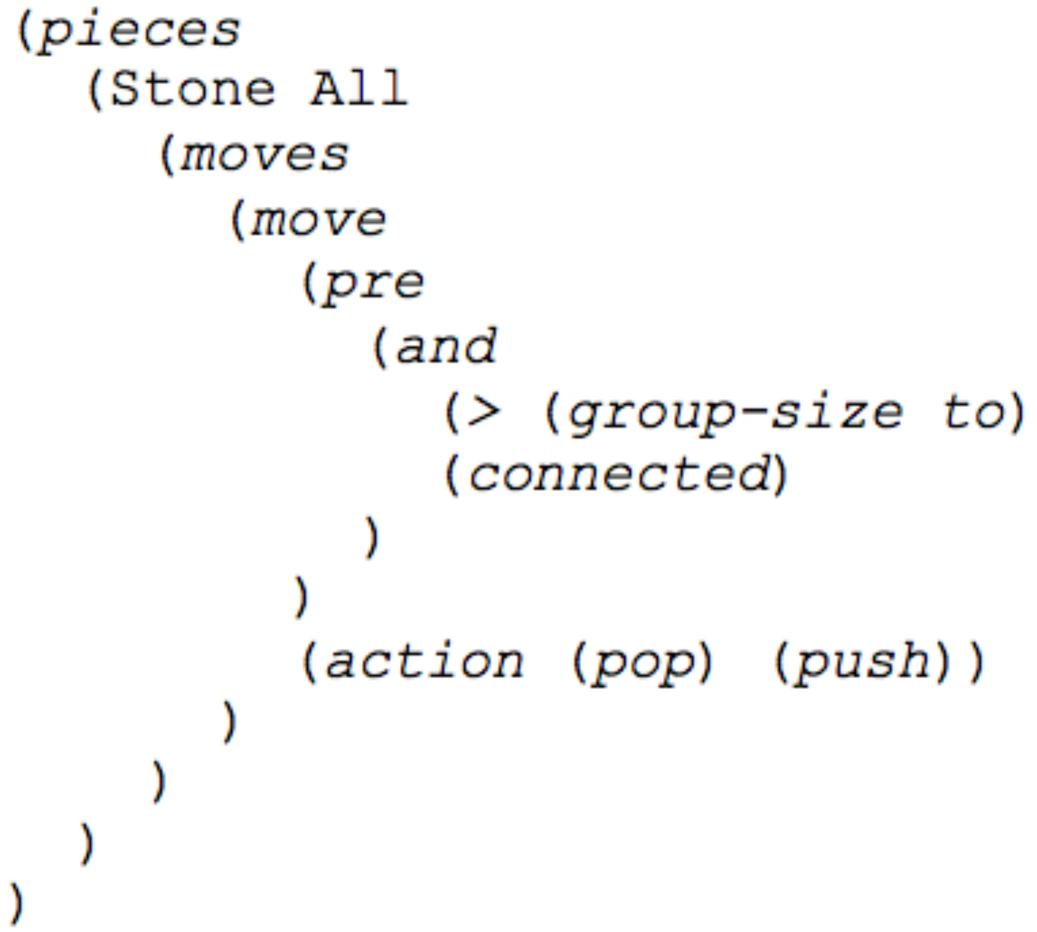
Yavalath was published by Browne and Nestor Games, and has a 7.2/10 rating on BoardGameGeek

Like METAGAME, Ludi uses a custom *game description language*.



```
(game Yavalath
  (players White Black)
  (board (tiling hex) (shape hex) (size 5))
  (end
    (All win (in-a-row 4))
    (All lose (and (in-a-row 3) (not (in-a-row 4))))
  )
)
```

Ludi has a little more control over the high-level game structure than METAGAME.



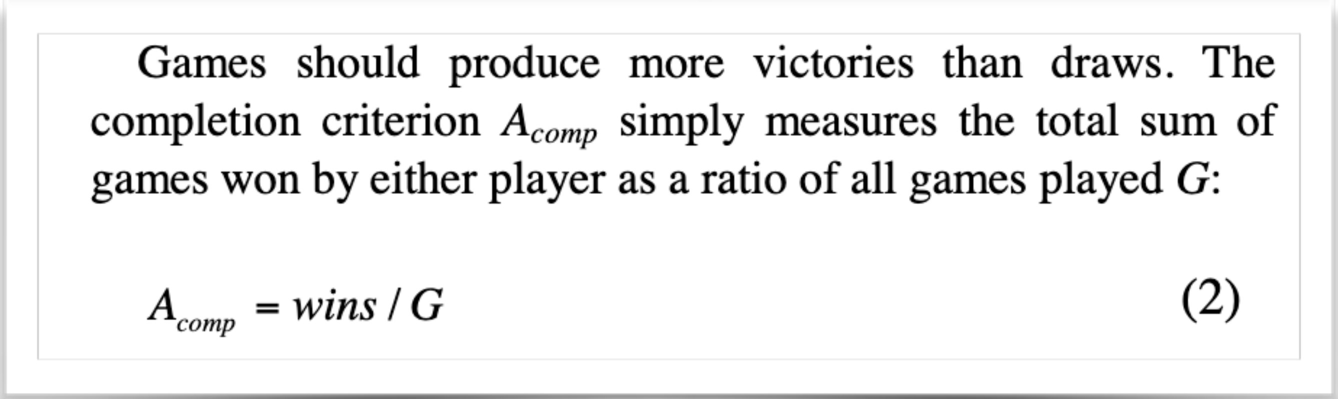
```
(pieces
  (Stone All
    (moves
      (move
        (pre
          (and
            (> (group-size to)
            (connected)
          )
        )
        (action (pop) (push))
      )
    )
  )
)
```

Rules are still mostly contained within pieces, and use a lot of predefined concepts (e.g. *group-size* or *connected*).

Browne and Maire’s main contribution is in **evaluation**.

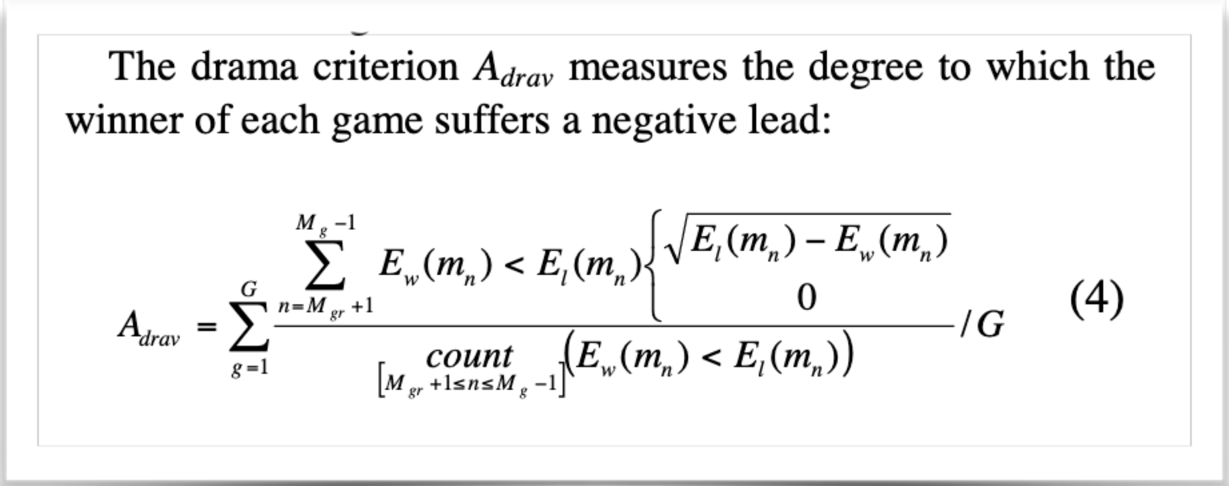
Ludi was able to play any game it could describe, meaning that it could test games and play them against human opponents.

This also meant it could potentially test games to see how good they were - but it needed something to measure. Browne and Maire supplied 57 (highly subjective) aesthetic criteria.



Games should produce more victories than draws. The completion criterion A_{comp} simply measures the total sum of games won by either player as a ratio of all games played G:

$$A_{comp} = wins / G \tag{2}$$



The drama criterion A_{drav} measures the degree to which the winner of each game suffers a negative lead:

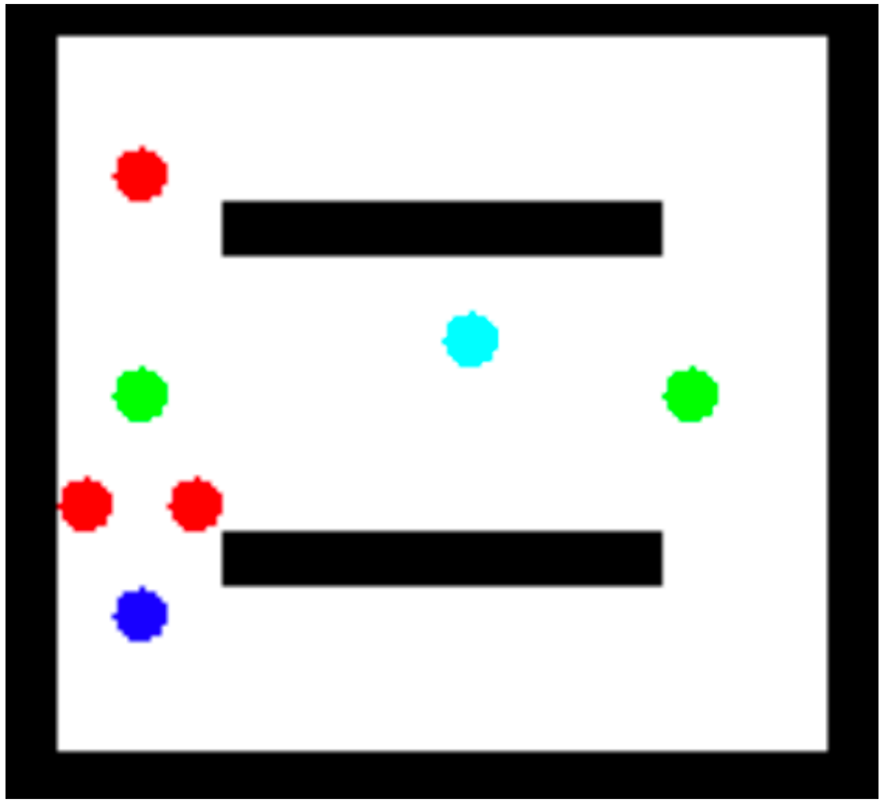
$$A_{drav} = \sum_{g=1}^{G} \frac{\sum_{n=M_{gr}+1}^{M_g-1} E_w(m_n) < E_l(m_n) \begin{cases} \sqrt{E_l(m_n) - E_w(m_n)} \\ 0 \end{cases}}{\underset{[M_{gr}+1 \le n \le M_g-1]}{count} \left(E_w(m_n) < E_l(m_n)\right)} / G \tag{4}$$

e.g. **Viability** - how many games ended in a draw

e.g. **Drama** - how many times was the eventual winner of the game in a losing position

Togelius & Schmidhuber

Togelius, J and Schmidhuber, J.
An Experiment in Automatic Game Design (2010)

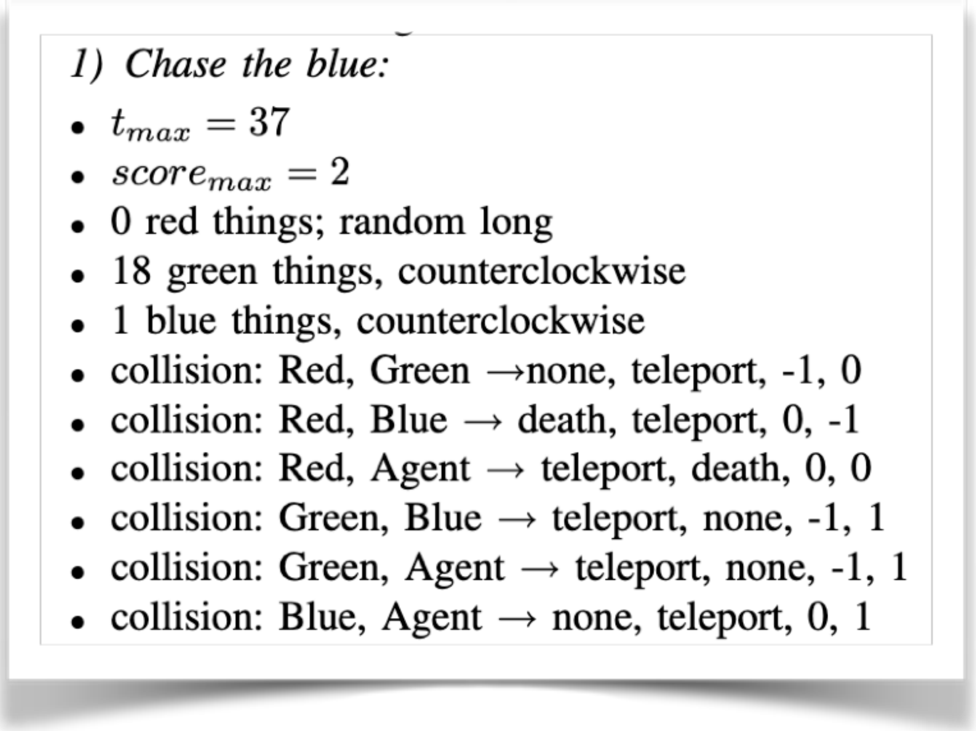


The example game in TS' paper.

Describes a system which generates simple arcade-like games, with a focus on evaluation.

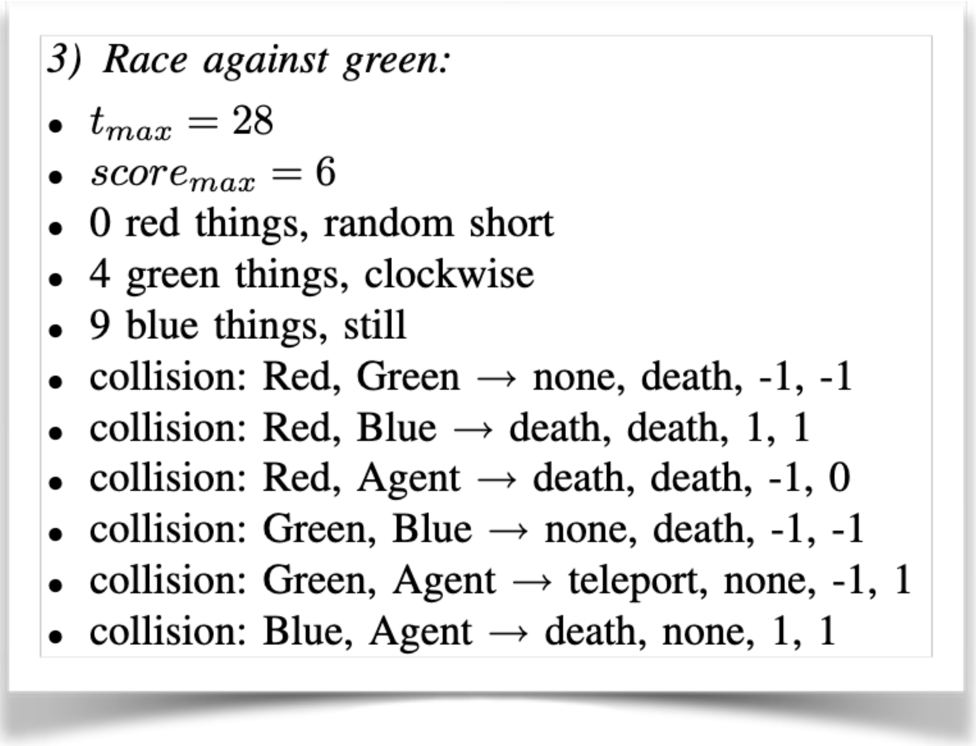
One of the first papers to automated digital game design. Notable for using neural networks to evaluate generated games.

Big Idea: Learning As Evaluation



1) Chase the blue:

- $t_{max} = 37$
- $score_{max} = 2$
- 0 red things; random long
- 18 green things, counterclockwise
- 1 blue things, counterclockwise
- collision: Red, Green →none, teleport, -1, 0
- collision: Red, Blue → death, teleport, 0, -1
- collision: Red, Agent → teleport, death, 0, 0
- collision: Green, Blue → teleport, none, -1, 1
- collision: Green, Agent → teleport, none, -1, 1
- collision: Blue, Agent → none, teleport, 0, 1



3) Race against green:

- $t_{max} = 28$
- $score_{max} = 6$
- 0 red things, random short
- 4 green things, clockwise
- 9 blue things, still
- collision: Red, Green → none, death, -1, -1
- collision: Red, Blue → death, death, 1, 1
- collision: Red, Agent → death, death, -1, 0
- collision: Green, Blue → none, death, -1, -1
- collision: Green, Agent → teleport, none, -1, 1
- collision: Blue, Agent → death, none, 1, 1

Ludi played games to evaluate them, using Browne and Maire's heuristics about what they thought made a good game.

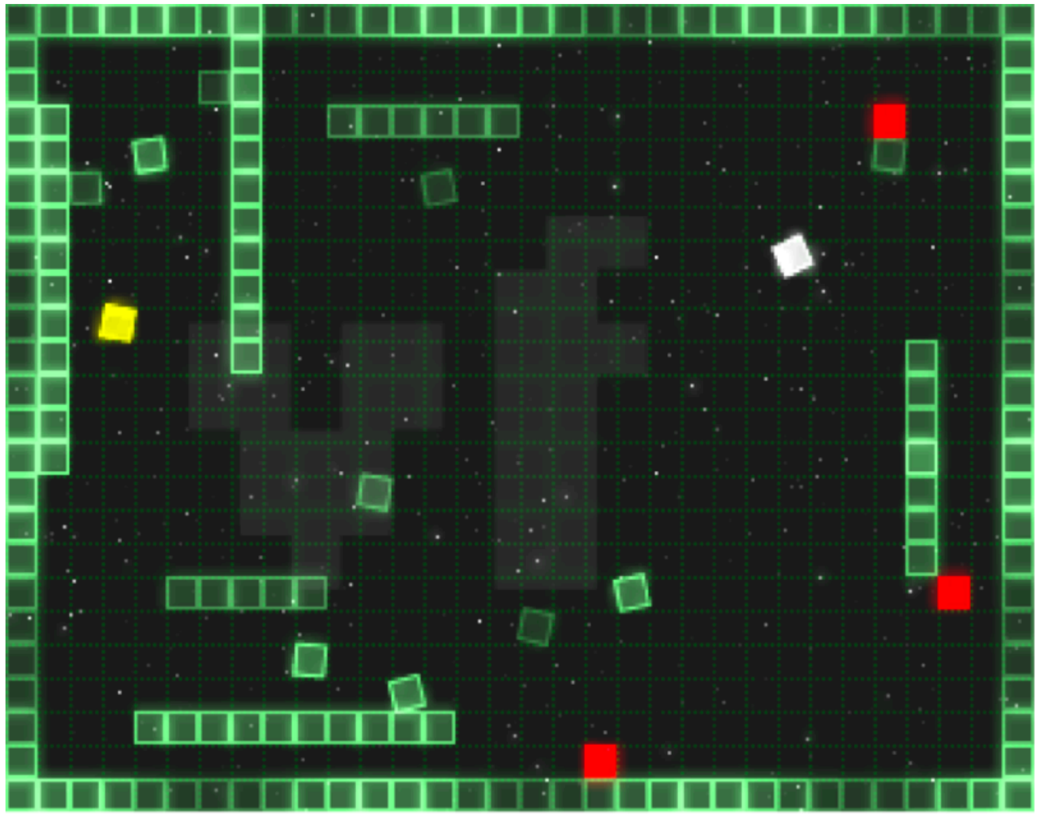
TS lets bots learn how to play the game, and measures how fast and how well they learn. Aim: not too fast, not too slow.

Although less explicit, this is another example of AGD researchers encoding their beliefs about game design in the systems they build.

"Learning is fun"

Variations Forever

Smith, A and Mateas, M. *Variations Forever - Flexibly Generating Rulesets From A Sculptable Design Space of Mini-Games (2010)*



Variations Forever

A game-as-generator which remixes rules in real time as the player plays.

Adam Smith is a huge proponent of *answer set programming*, which allows for rapid solving of design problems and can elegantly let a designer specify constraints.

Automated Game Design

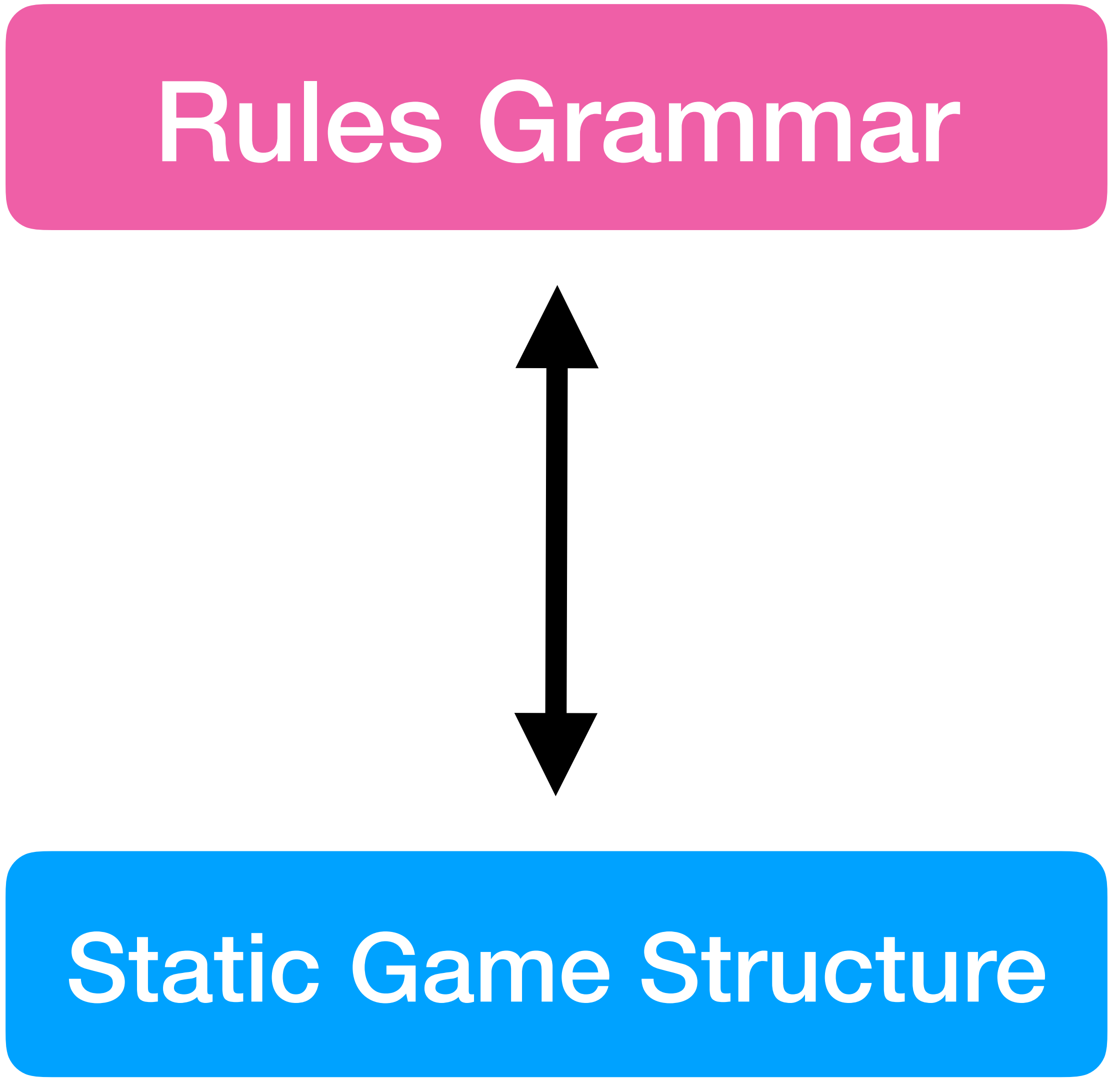
1st Generation

Games = Rules

Game Design = Rule Design

Automated Game Design

1st Generation



Rules Grammar

Static Game Structure

AI has a long history with games-as-rules

Thinking about games had developed a lot since the 50s

AGD & Proceduralism

In 2005, Michael Mateas wrote a paper calling for a greater emphasis on proceduralism in new media education.

> ***“Code is a kind of writing… new media scholars must read code… at the level of the procedural rhetoric, aesthetics and poetics encoded in a work.”***

Ian Bogost also wrote about the potential for procedural rhetoric, especially in the design and analysis of **persuasive games**.

AGD & Proceduralism

Interest in proceduralism had a large (and controversial) impact on games studies, but it also impacted game AI.

In 2006 Mateas co-founded the Expressive Intelligence Studio at UC Santa Cruz, now a hugely influential institution in game AI.

A new wave of automated game design projects came out of UC Santa Cruz over the following years, influenced by proceduralism.

As game AI research grew, it was also developing a richer understanding of what games were, and could be.

1st Generation

1992-2010

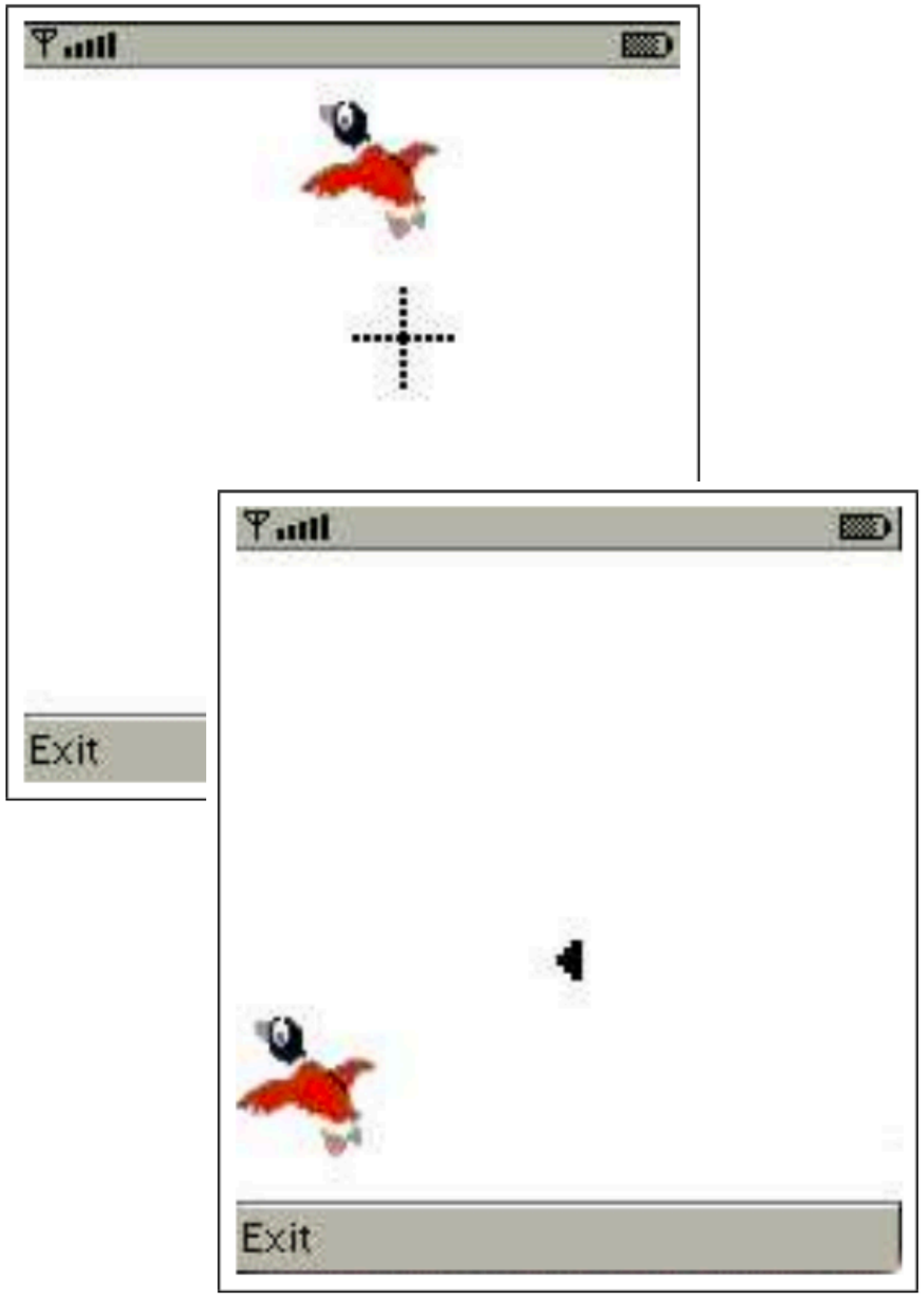
2nd Generation

2010-2016

3rd Generation

2016-????

Nelson & Mateas

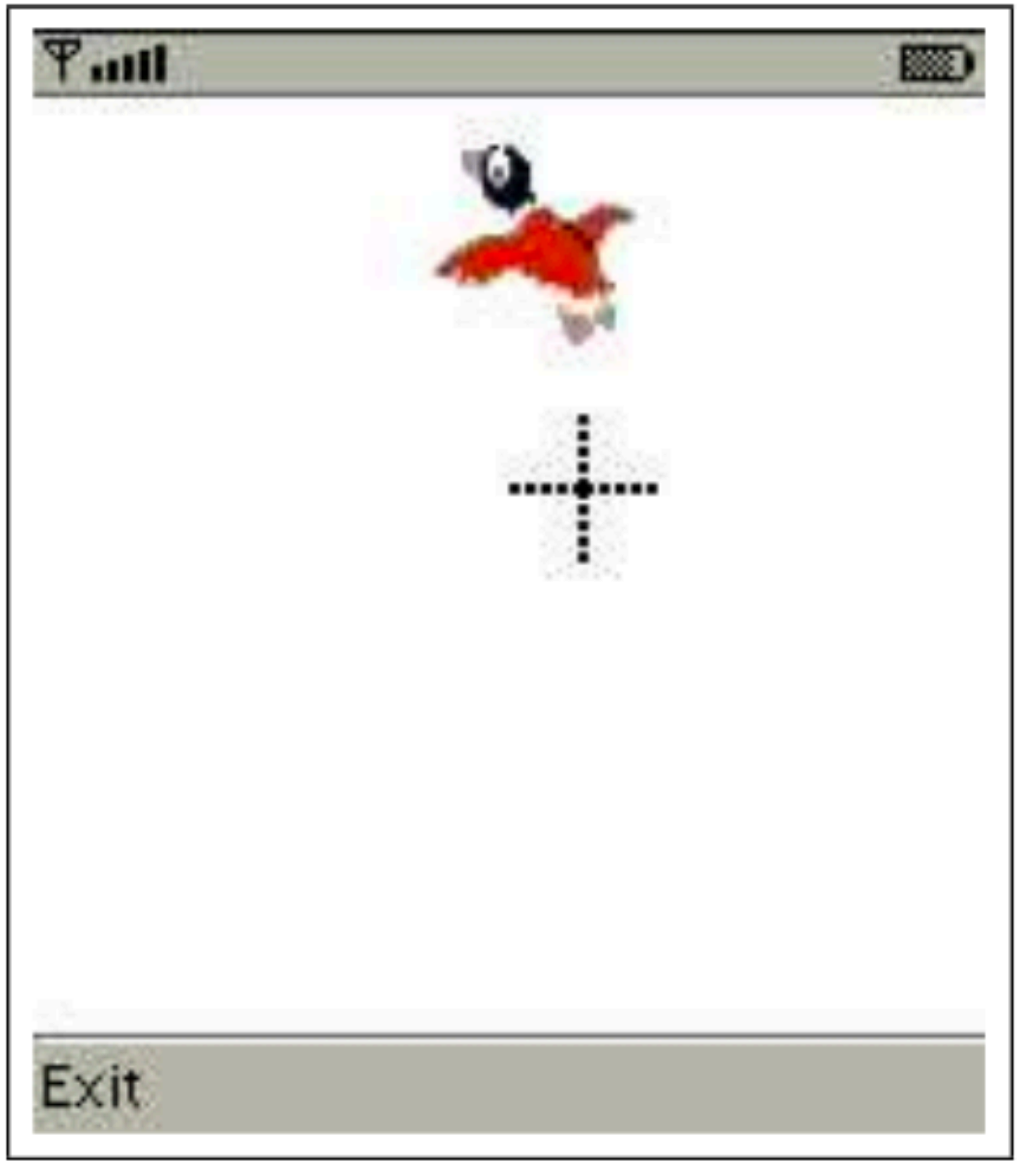


Nelson, M and Mateas, M. *Towards Automated Game Design* (2009)

Nelson and Mateas describe a system which can generate a simple mini-game from a two-word prompt.

The earliest example of game generation which incorporated thematic and visual elements.

Example: Shooting Pheasants

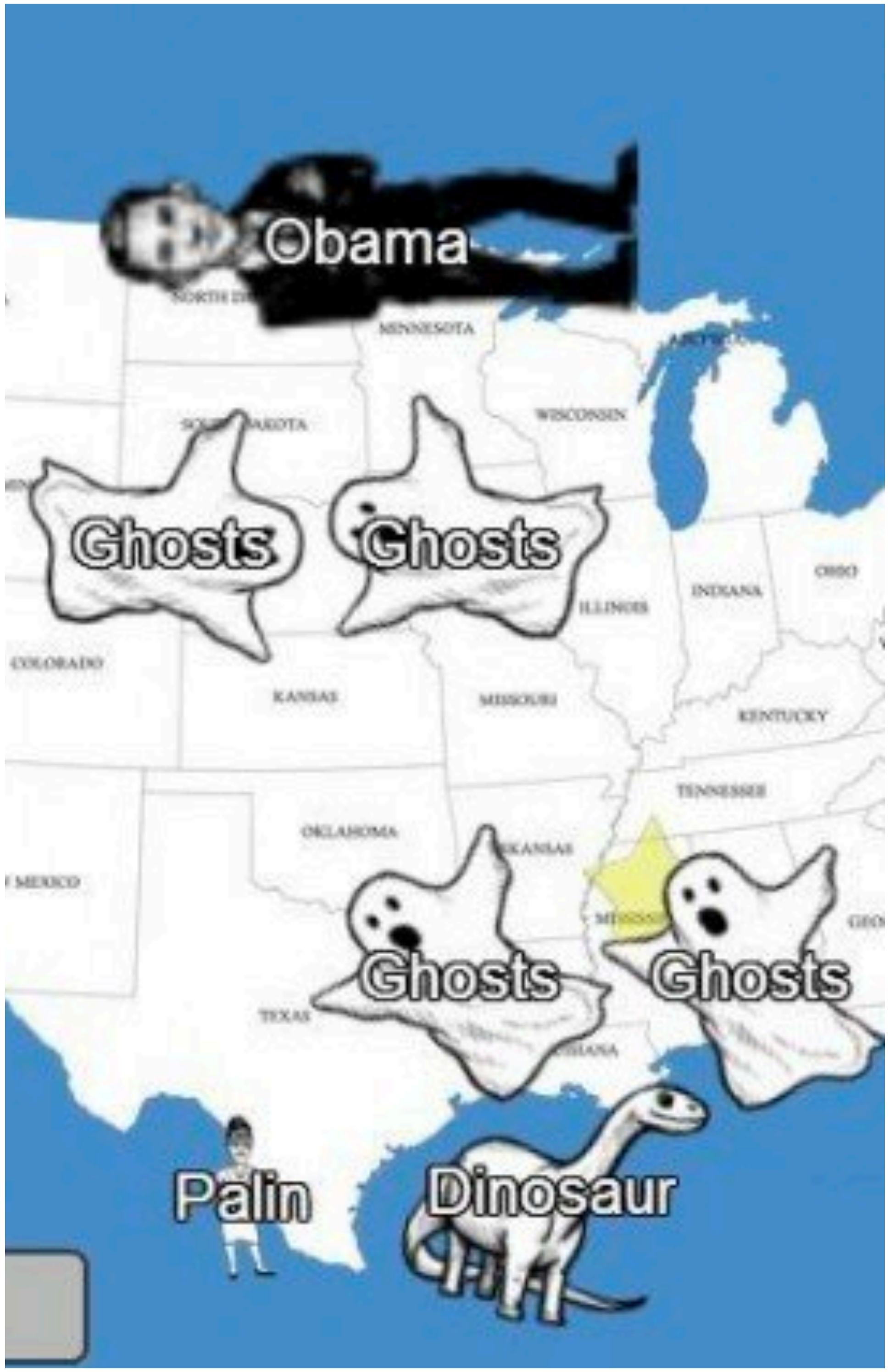


The system has a limited vocabulary, but can link words to concepts it knows, so *Pheasants* becomes *Ducks*.

The system can produce multiple games for a single input phrase. e.g. a game where you shoot ducks, or a game where you control a duck and avoid being shot.

Despite being a decade old, these games still work on modern hardware, thanks to some clever planning!

Game-O-Matic

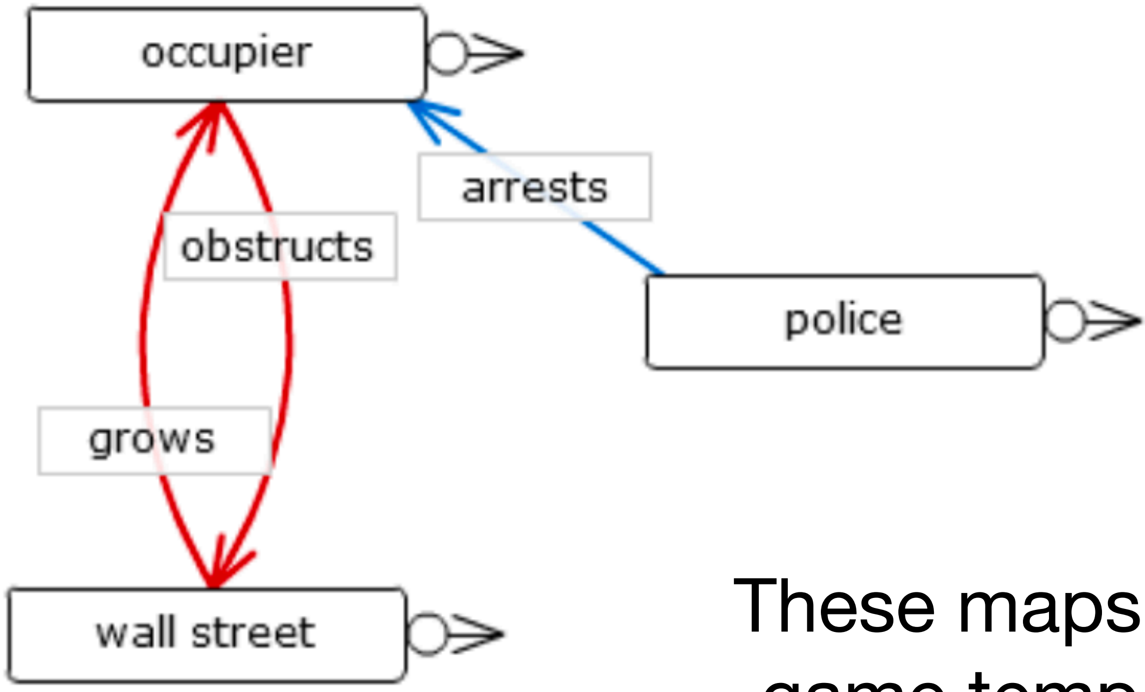


Treanor, Blackford, Mateas and Bogost. *Game-o-Matic* (2012)

The Game-o-Matic was an assistive automated game designer that was aimed at journalists and the public.

The GoM works from *concept maps* which enable it to make richer games that connect to real-world ideas.

Example: Occupy Wall Street



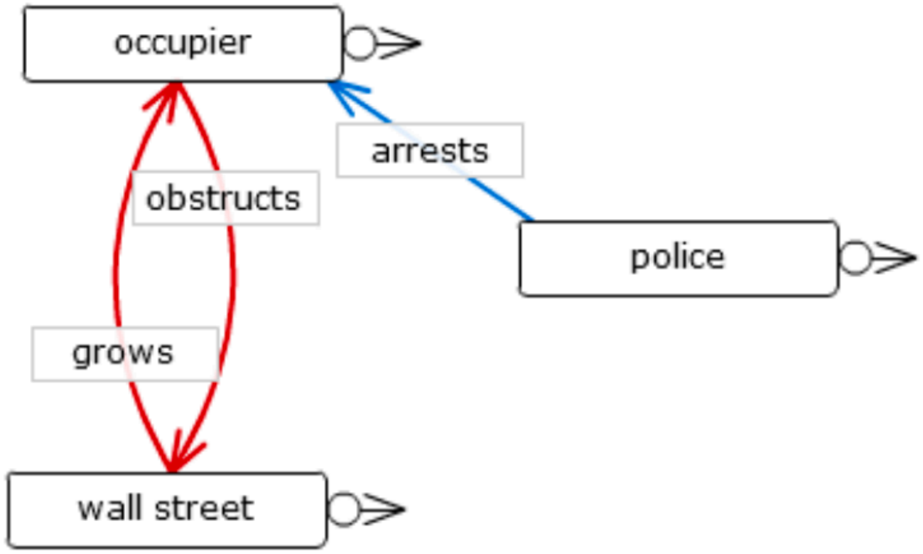
Conceptual maps define game pieces (like *protester* or *police*) and the relationships between them (like *arrests*).

These maps can be quite complex, so simple game templates won't work. The GoM uses *recipes* and *micro-rhetorics* to find an appropriate game design.

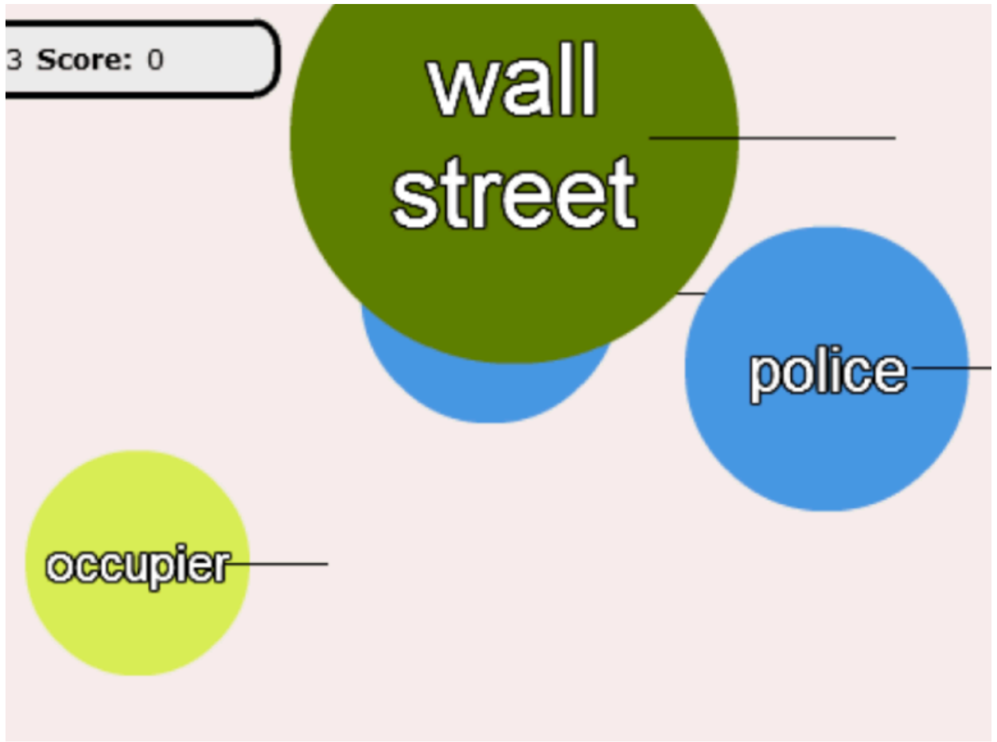
Like NM, the GoM can design games from different perspectives, and works from a fixed list of known game mechanics.

Micro-Rhetorics

Treanor wrote a catalogue of **micro-rhetorics** which related rules to meaning. e.g. if A touches B and B disappears, this might be interpreted as A eating B.

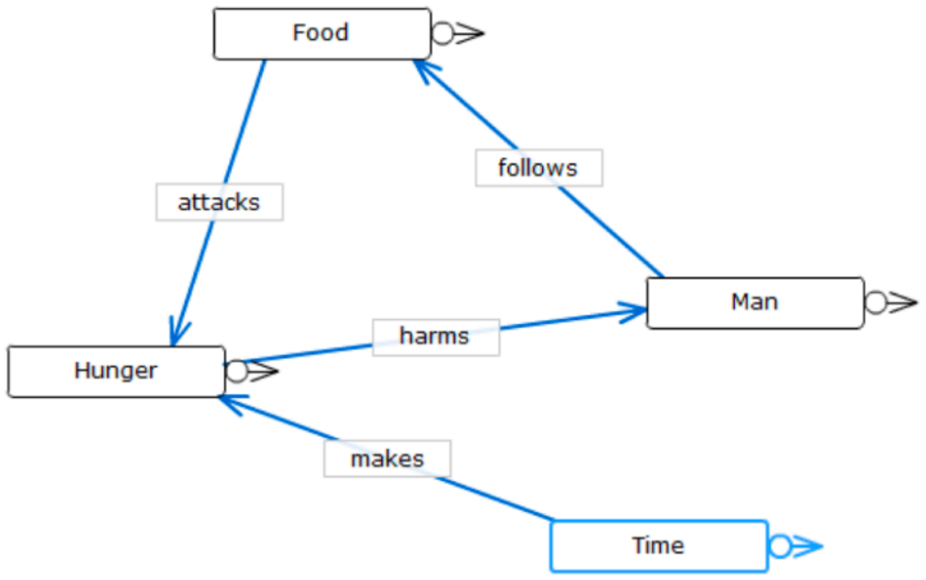


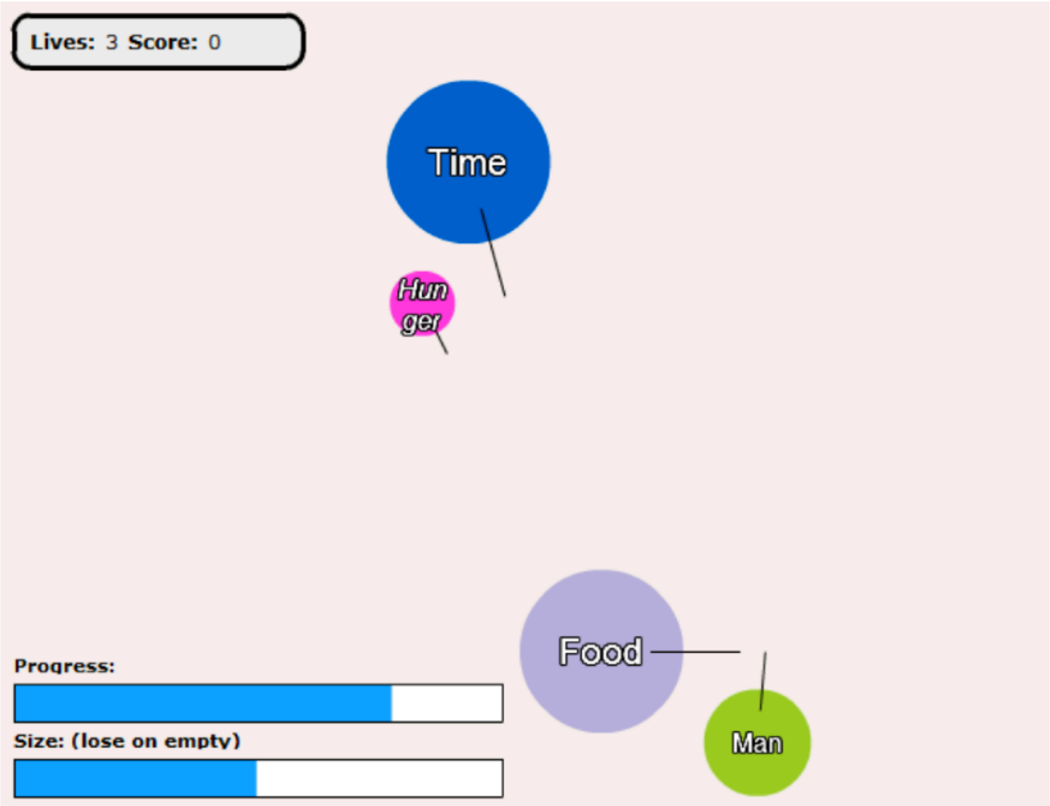
This allowed the GoM to link conceptual maps (*police arrest protesters*) to game mechanics (*A chases and destroys B).*



Mateas describes the GoM as “an embodied theory of newsgames”. This is a common theme in UCSC projects - building systems that embody theories about game design.

Recipes





Micro-rhetorics can be combined into full games using *recipes* which were closer to full game templates.

For example, if A wants to destroy B, and C wants to protect B, a recipe might describe a game where the player is A and has to chase B while avoiding C.

These all require a fair amount of human input, and a lot of the information is culturally relative or subjective.

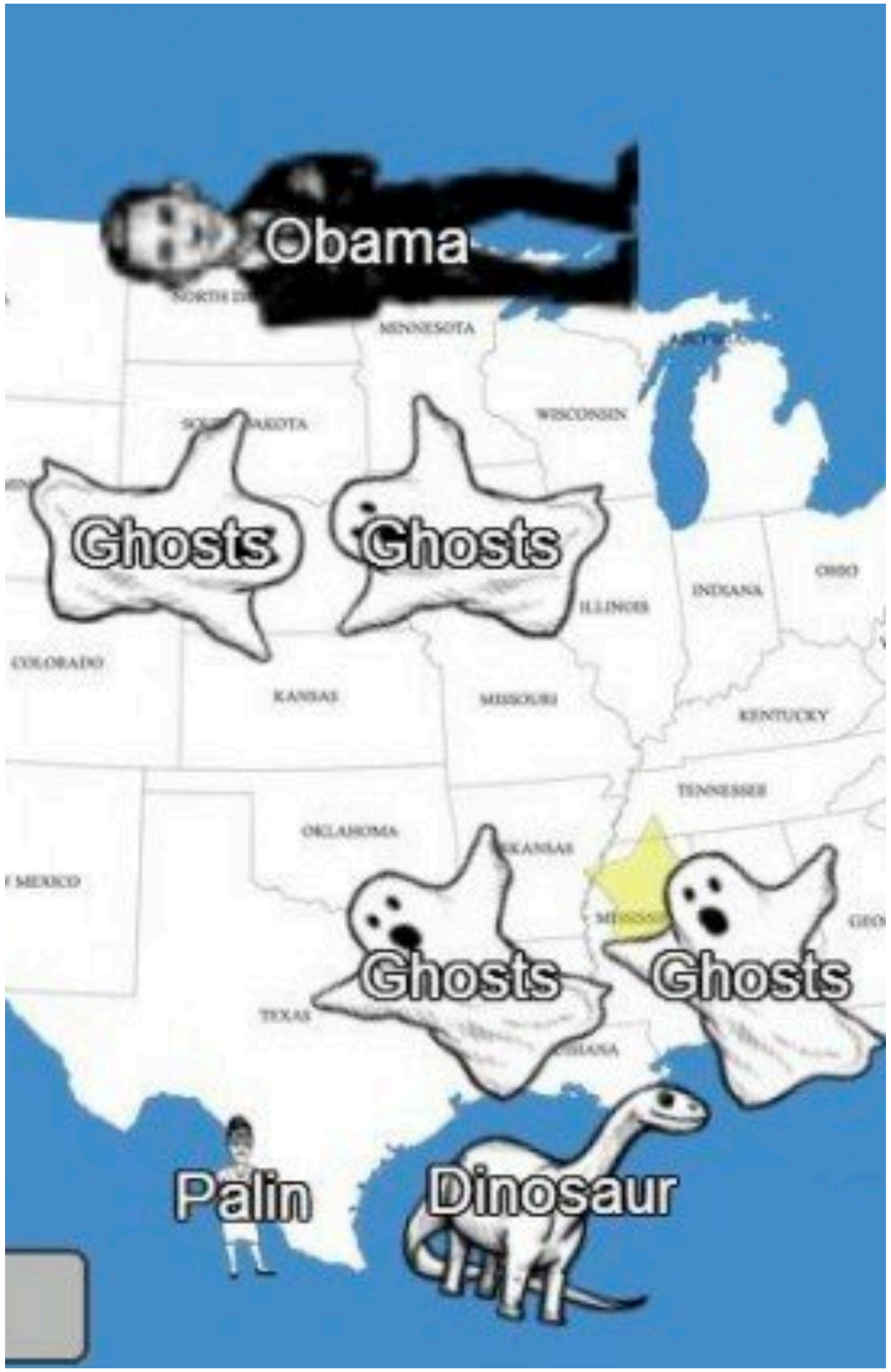
AGD & Proceduralism

Both NM and the GoM show elements of how proceduralism was changing the course of AI in game design.

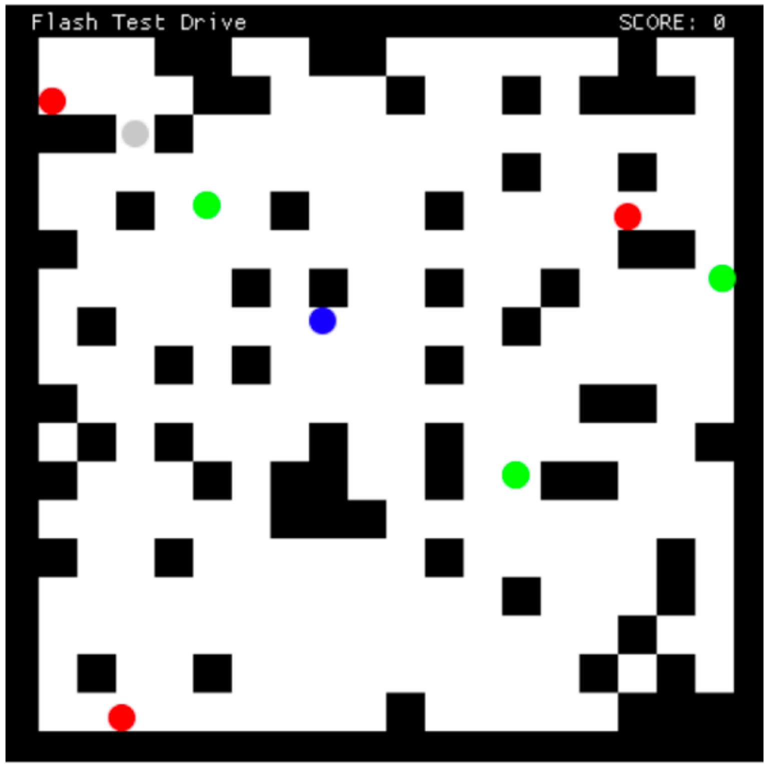
Proceduralism was appealing because it established a direct link between the technical and the artistic in games.

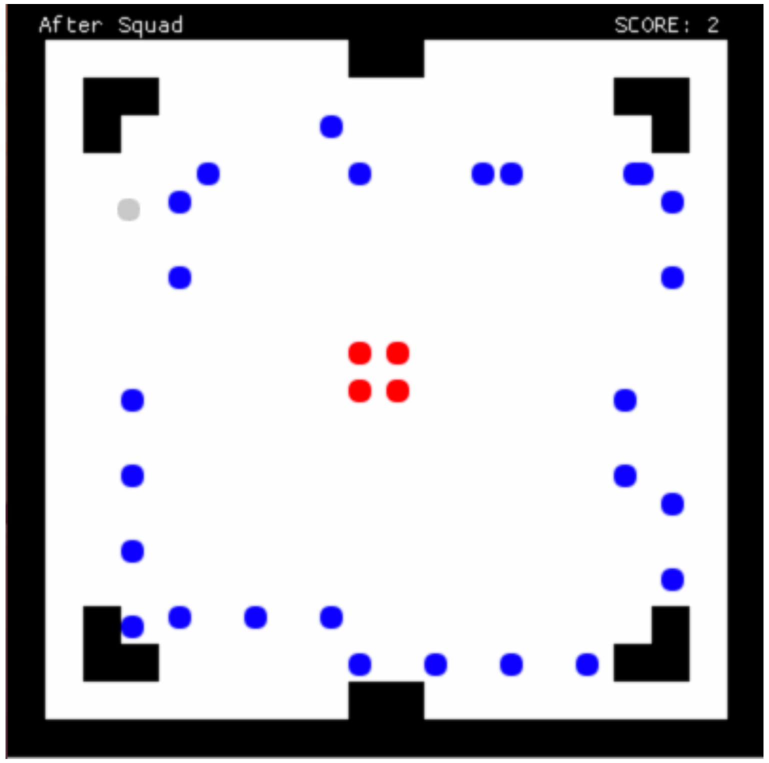
Critiques of procedurality in game studies argued that it was too simplistic and overly reductive, and it remains divisive.

Despite these problems, proceduralism was a driving force in bringing AGD out of abstract-only games and beginning to consider elements like **meaning**.



ANGELINA





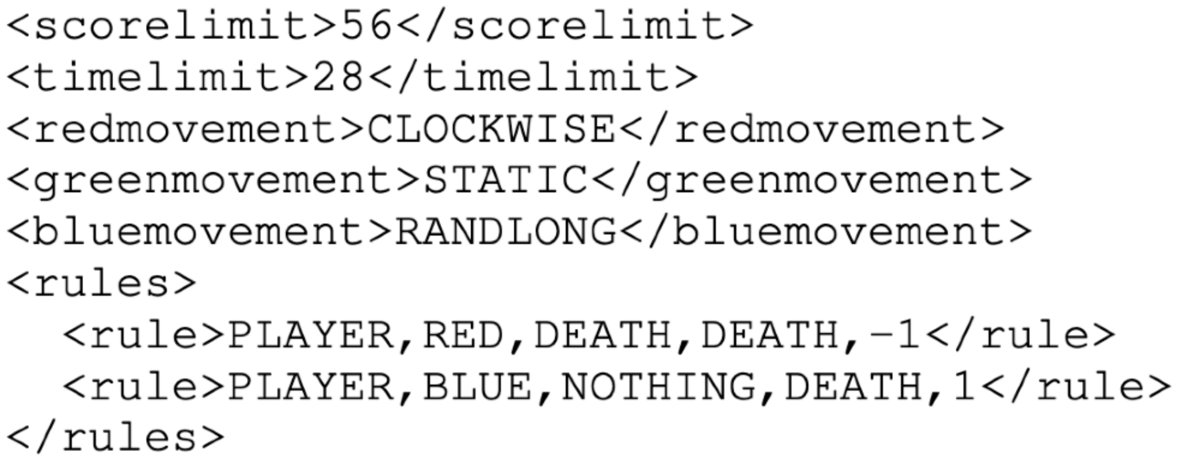
Cook, M and Colton, S. *Multi-Faceted Evolution Of Simple Arcade Games* (2011)

The first version of an automated game designer. Draws both from Game AI research and Computational Creativity.

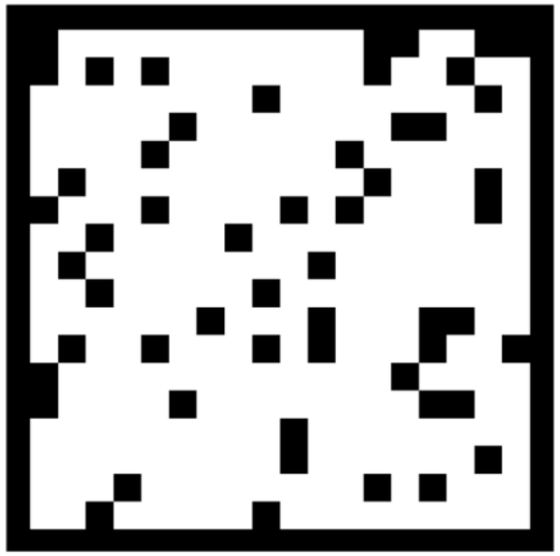
This version uses a domain inspired by Togelius & Schmidhuber, but designs more of the game itself, and does all aspects simultaneously.

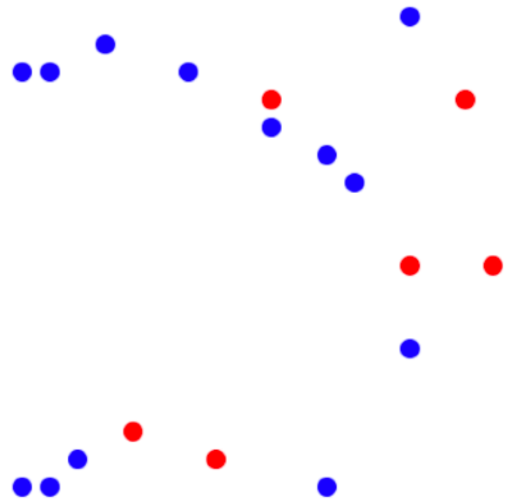
ANGELINA

An important emphasis is that no part of the game is more important than the others: **de-emphasising rules** in the process and **adding new design tasks** to the problem.



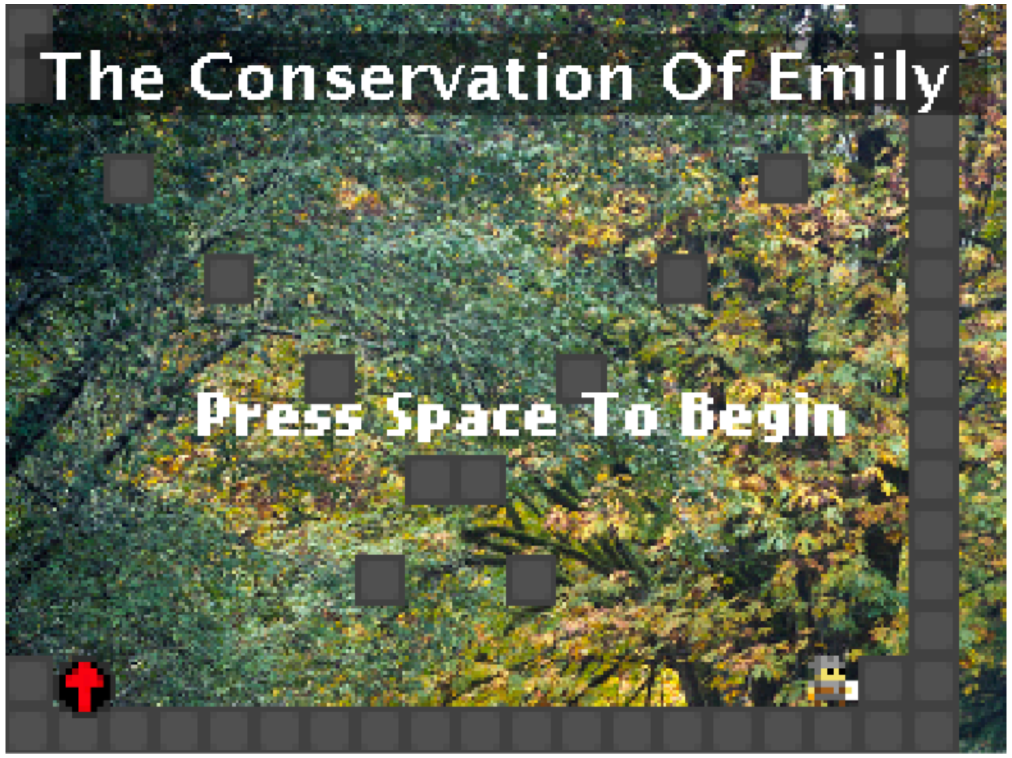
```
<scorelimit>56</scorelimit>
<timelimit>28</timelimit>
<redmovement>CLOCKWISE</redmovement>
<greenmovement>STATIC</greenmovement>
<bluemovement>RANDLONG</bluemovement>
<rules>
  <rule>PLAYER,RED,DEATH,DEATH,-1</rule>
  <rule>PLAYER,BLUE,NOTHING,DEATH,1</rule>
</rules>
```

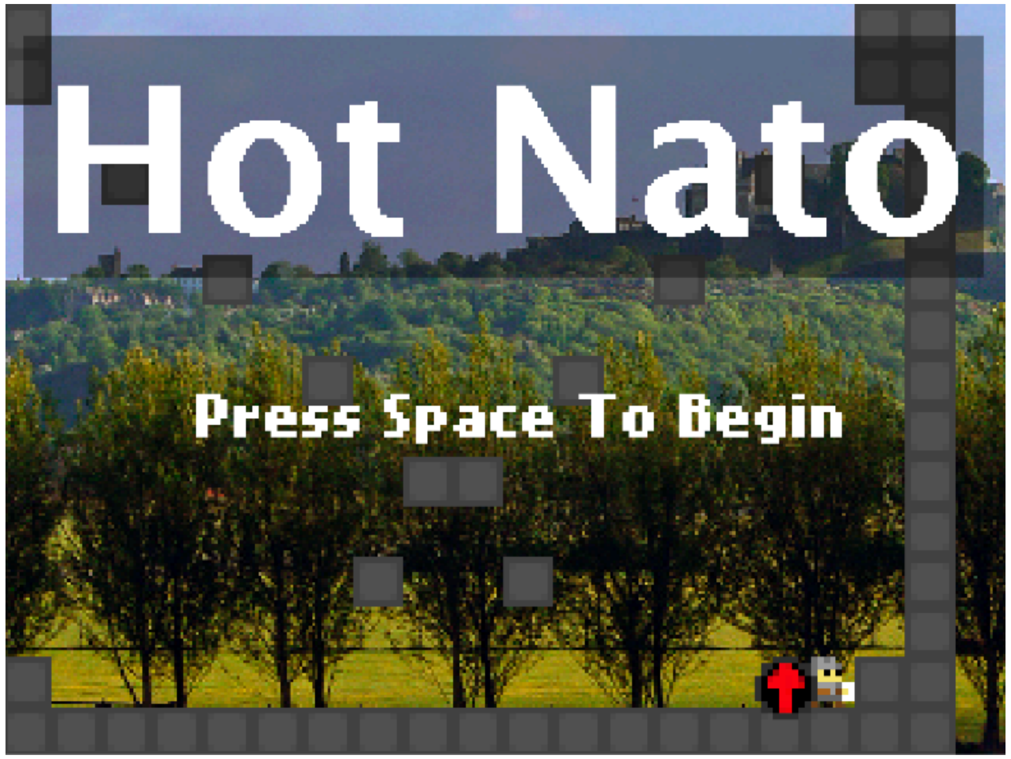




Successive versions of the system add more design tasks to the problem, which was partly inspired by the Computational Creativity interest in ‘handing over responsibility’ to the AI.

ANGELINA v3



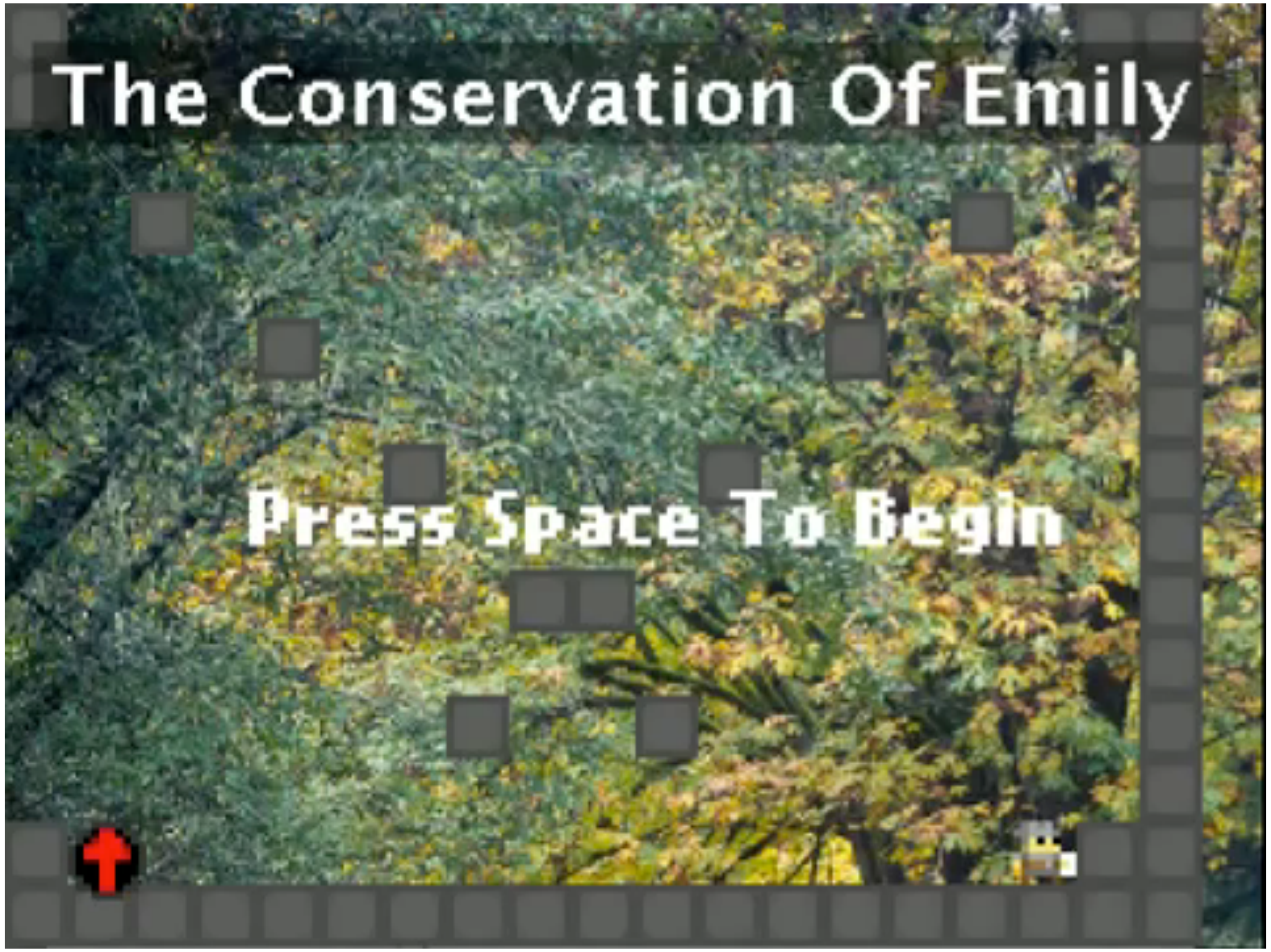


Cook, M, Colton, S and Pease, A. *Aesthetic Considerations for Automated Platformer Design* (2013)

Version 2 of ANGELINA laid groundwork for platformer generation, with more granular rules, but version 3 added in thematic/artistic control for the first time.

Allowed ANGELINA to make pseudo-newsgames, inspired by news articles, but not intended to persuade or argue (like the Game-O-Matic).

Example Game: The Conservation of Emily



The most impactful features of this version are the attempts to link the game to a real-world topic.

Images, sound effects, music were related to the topic, and game names were ‘clever’ puns.

However, this began to test the limits of computational evolution. A lot of the most interesting parts of the game design were done outside of the evolutionary loop.

Away From Grammars

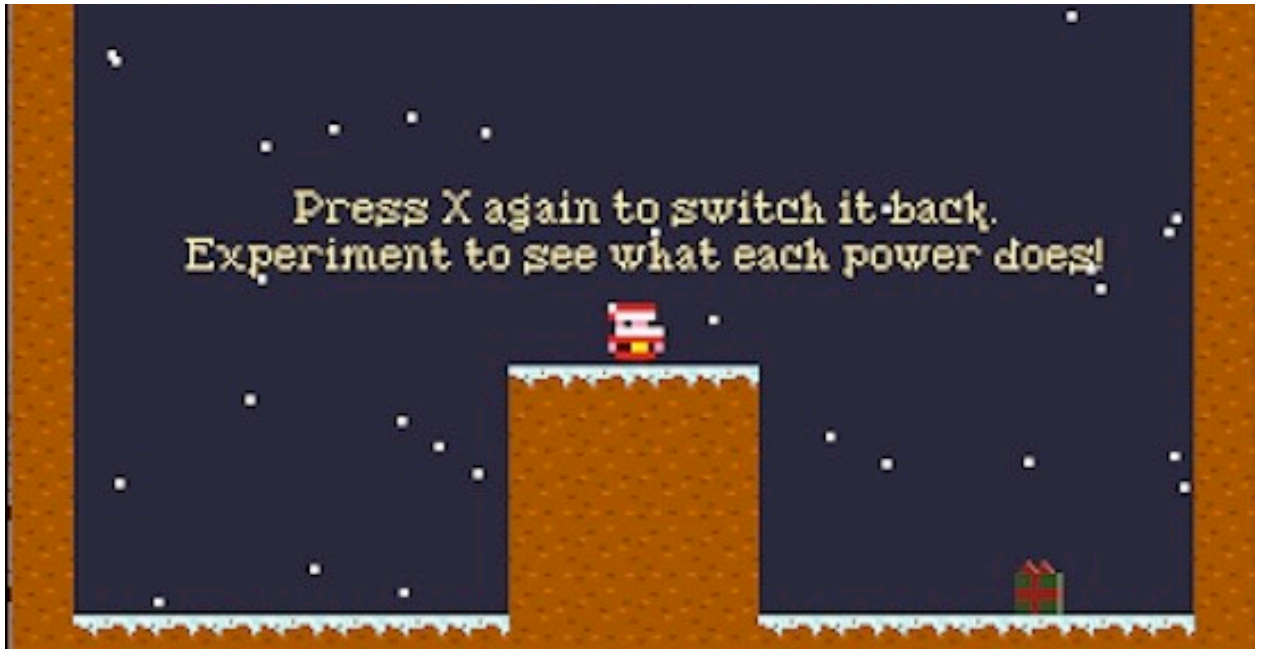
```
effect = DOUBLEJUMP();
```

Previous systems all relied heavily on grammars, allowing people to embed design knowledge.

```
player.jump += 150;
```

ANGELINA used some embedded knowledge, but could also edit the code to fine-tune variables.

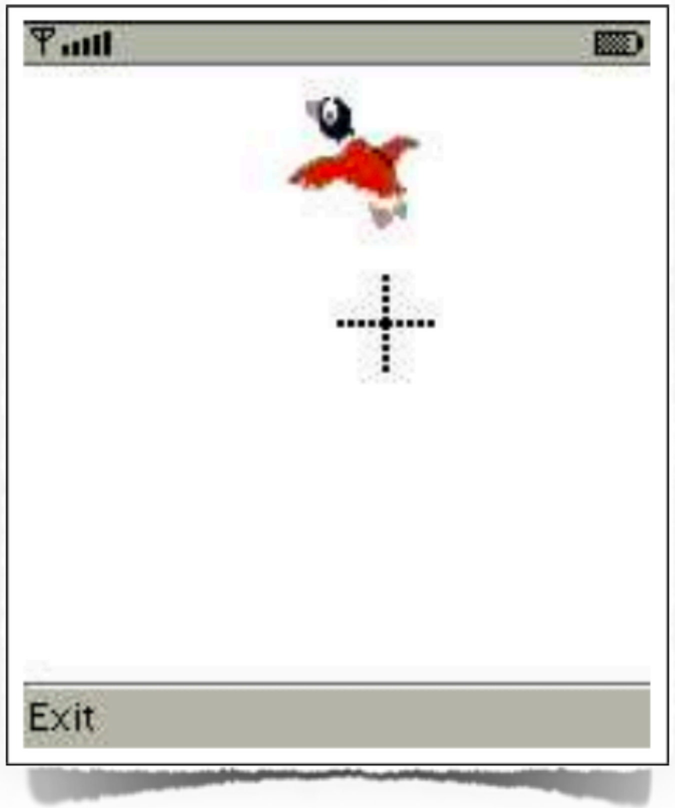
We expanded on this later in a system called **Mechanic Miner**, which generated code directly (beyond the scope of this lecture)

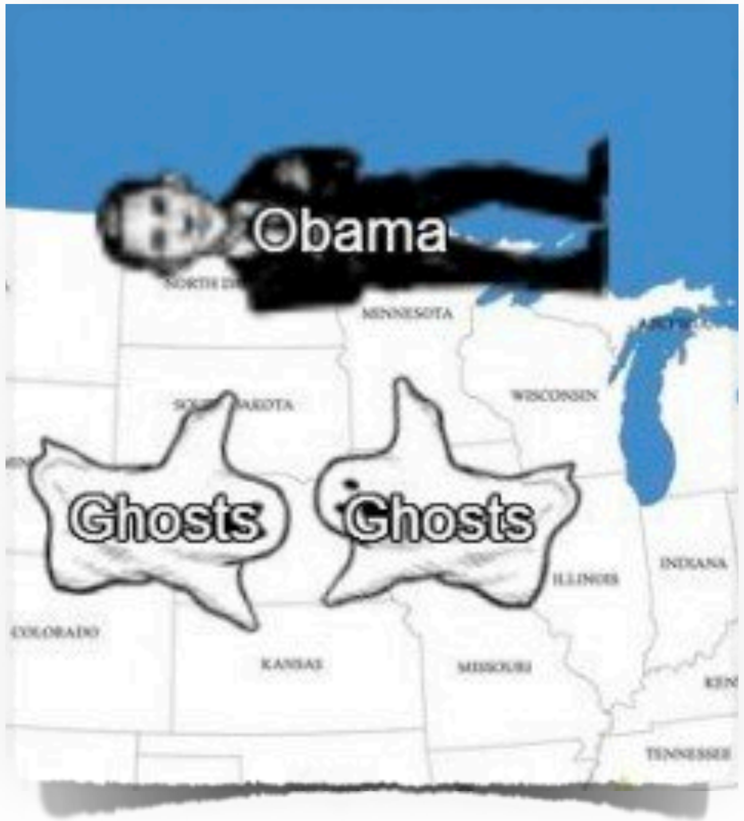


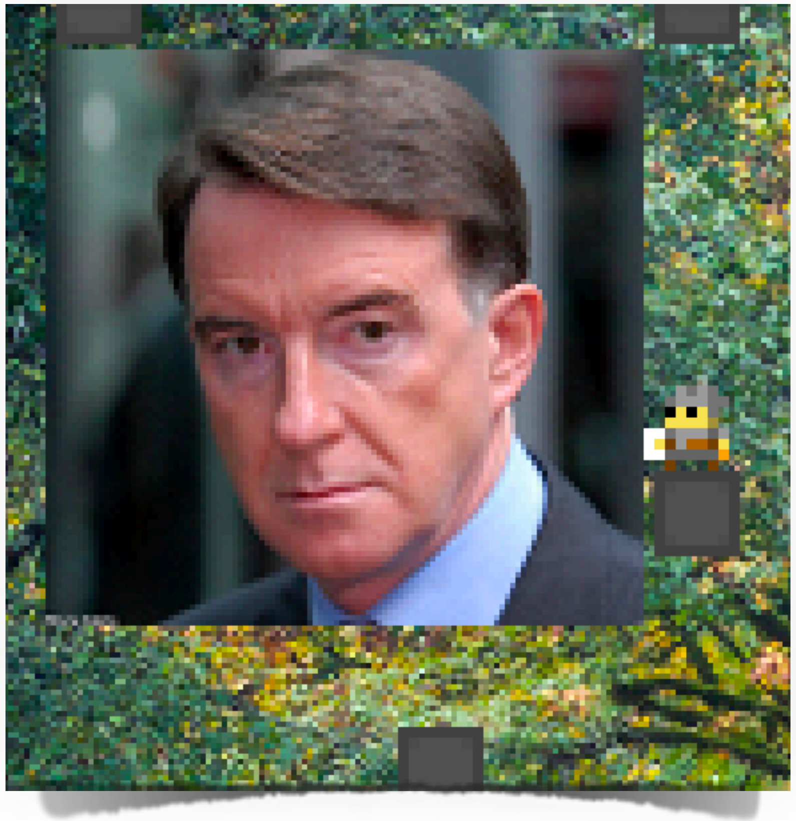
Press X again to switch it back.
Experiment to see what each power does!

Context Is Everything

Av3 designed four game components: **rules** (item progression); **levels** (raw geometry); **challenges** (enemy and item placement) and **art** (including background images, music and title).



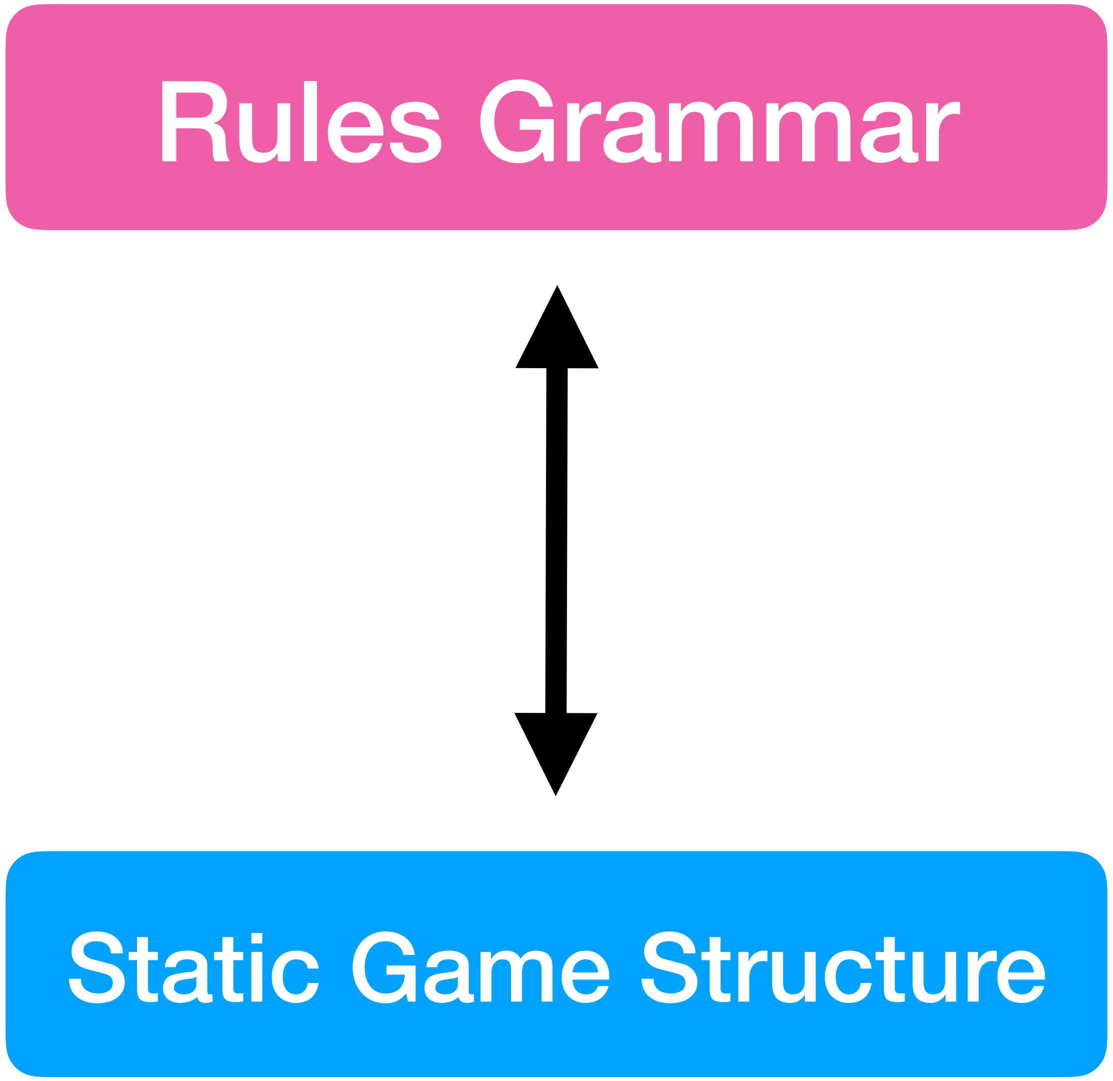




A key trend throughout this generation of AGD systems is an expansion beyond rules, to consider the other elements of a game and their importance - particularly **meaning**.

Automated Game Design

1st Generation

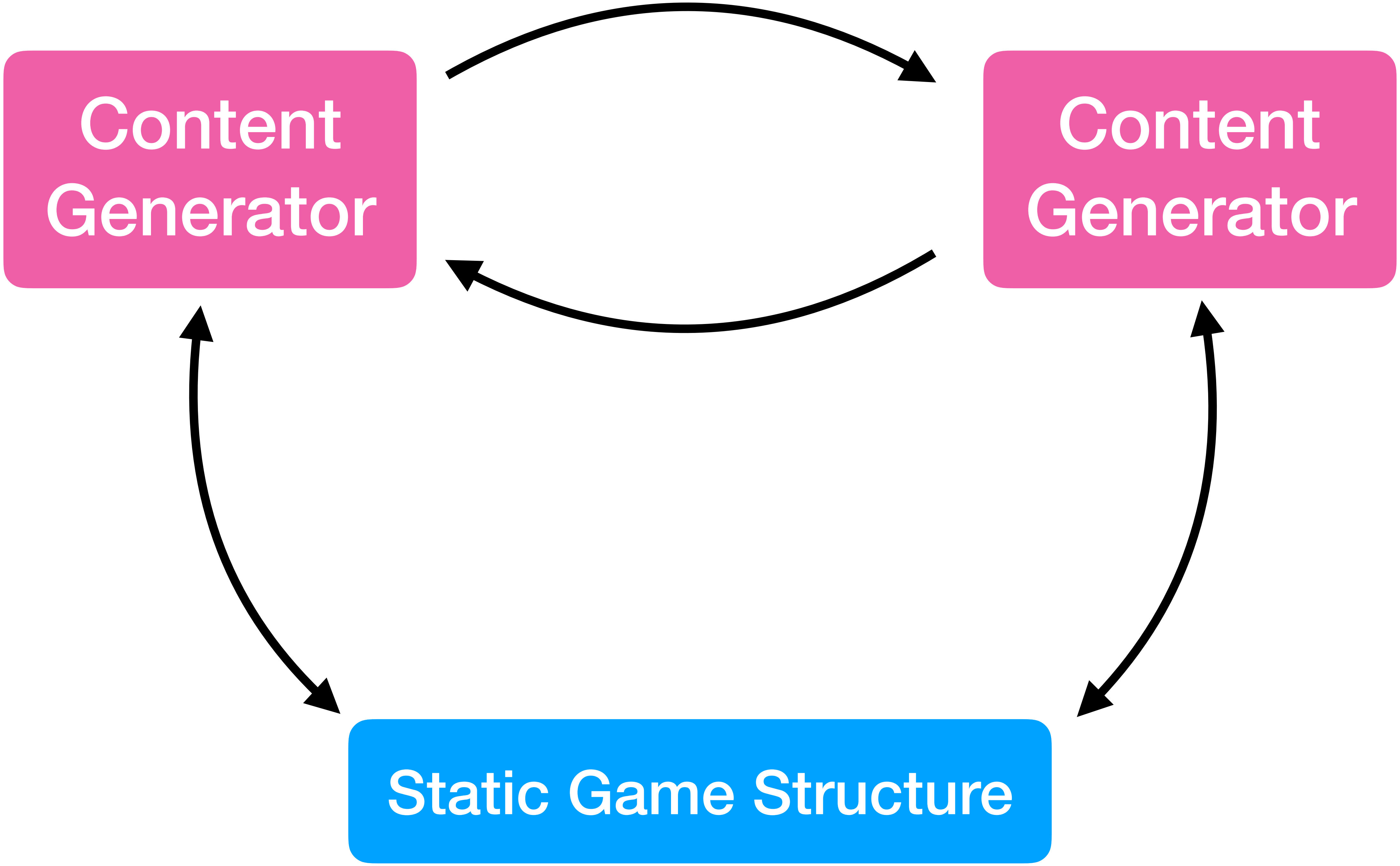


Rules Grammar

Static Game Structure

Automated Game Design

2nd Generation



Content Generator

Content Generator

Static Game Structure

Automated Game Design

1st Generation

Games = Rules

Game Design = Rule Design

Automated Game Design

2nd Generation

Games = Rules, Art, Music, etc…

Automated game design is the process of generating ***two or more*** *types of content in a way that is* ***aware of the changing game design****.*

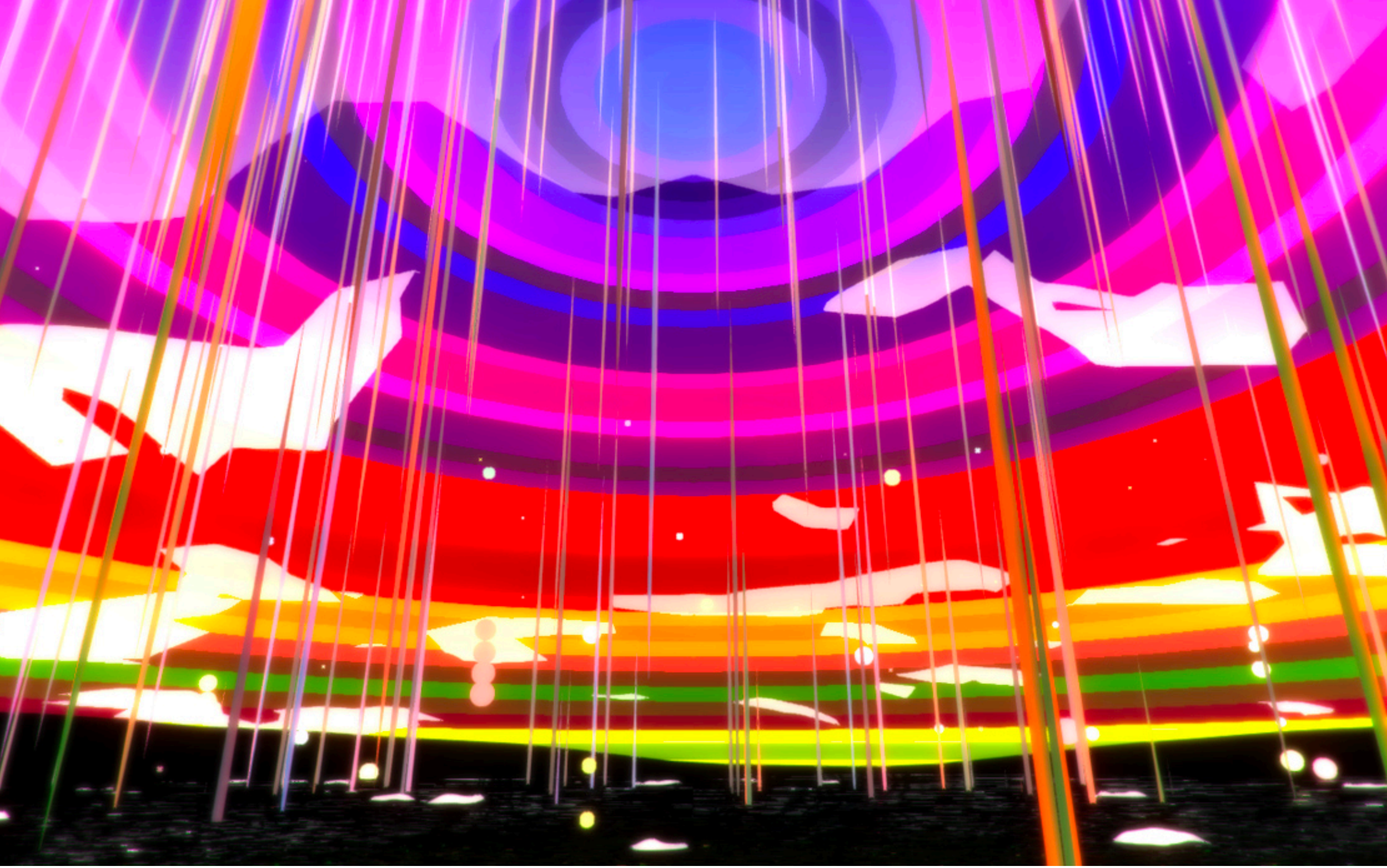
AGD & Orchestration

This idea of AGD as linked-up content generation was supported by Antonios Liapis et al in their 2014 paper.

> ***"The orchestration of these facets into a fully automatically generated game entity is a challenging future direction for [AI] research"***

Liapis, Yannakakis and Togelius. *Computational Game Creativity* (2014)

They proposed that automated game design was largely a challenge of **orchestrating** different aspects of a game.



Which bits of a game are important?

AGD & Formalism

I wrote a paper with Gillian Smith calling for AGD research to consider moving away from games centred around rules.

> ***"Most current AGD systems make the statement that games are a set of rules from which aesthetics emerge… [they] ignore the player, in favour of a formalist, structural, mechanics-privileged approach to game design."***

Cook and Smith. *Formalising Non-Formalism: Breaking The Rules of Automated Game Design (2015)*

Automated game design still struggles with this today - and even I have found it hard to break out of this entirely.

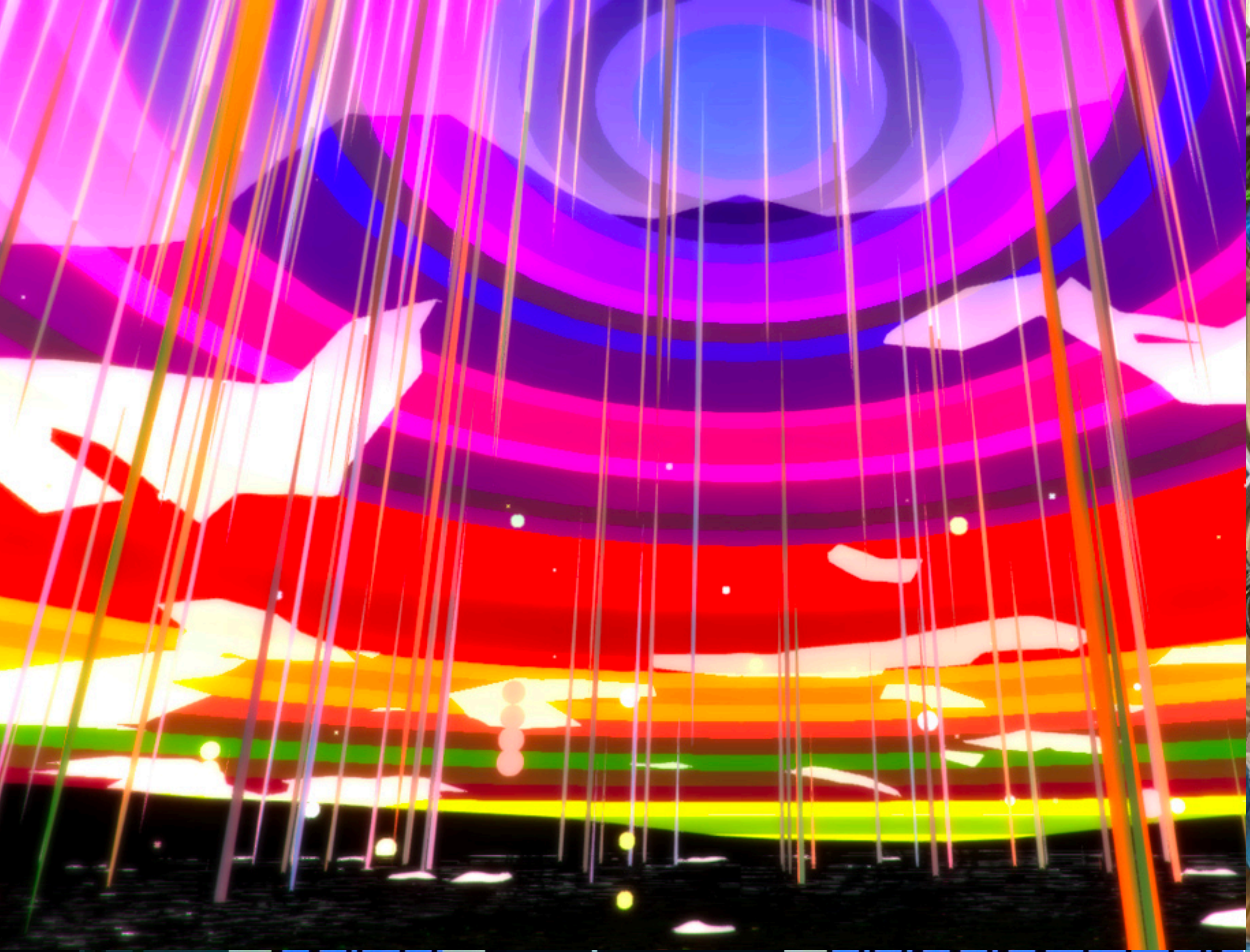
Games are many different things

- Art
- Sound
- Music
- Writing
- Systems
- Feel
- Metagame
- Interface
- Social
- Difficulty
- Discovery
- Self-Expression
- Juice
- Oil
- Progression
- Co-operation
- Culture
- Context
- Meaning
- Balance
- Emergence
- Subversion
- Expansion
- …

Games can be a lot of things

- Art
- Sound
- Music
- Writing
- Systems
- Feel
- Metagame
- Interface
- Social
- Difficulty
- Discovery
- Self-Expression
- Juice
- Oil
- Progression
- Co-operation
- Culture
- Context
- Meaning
- Balance
- Emergence
- Subversion
- Expansion
- …

AI tempts us into thinking maximally



Battlecry: Give a friendly minion Divine Shield.

- Art
- Sound
- Music
- Writing
- Systems
- Feel
- Metagame
- Interface
- Social
- Difficulty
- Discovery
- Self-Expression

- Juice
- Oil
- Progression
- Co-operation
- Culture
- Context
- Meaning
- Balance
- Emergence
- Subversion
- Expansion
- …

There’s no right answer

- Art
- Sound
- Music
- Writing
- Systems
- Feel
- Metagame
- Interface
- Social
- Difficulty
- Discovery
- Self-Expression

- Juice
- Oil
- Progression
- Co-operation
- Culture
- Context
- Meaning
- Balance
- Emergence
- Subversion
- Expansion
- …

There’s no right answer

- Art
- Sound
- Music
- Writing
- Systems
- Feel
- Metagame
- Interface
- Social
- Difficulty
- Discovery
- Self-Expression

- Juice
- Oil
- Progression
- Co-operation
- Culture
- Context
- Meaning
- Balance
- Emergence
- Subversion
- Expansion
- …

There’s no right answer

“Which future of games is correct? All of them.”

- - George Buckenham

A common link?

“Because it’s there”

Helping Developers

Helping Everyone

Changing Games

People

1st Generation

1992-2010

2nd Generation

2010-2016

3rd Generation

2016-????

AGD & Computational Creativity

In 2012, Simon Colton and Geraint Wiggins wrote a modern definition of computational creativity:

> ***"…the engineering of computational systems which… exhibit behaviours that unbiased observers would deem to be creative."***

In their view, creative tasks were impossible to evaluate objectively. There was no 'formula' or 'test' for creativity. Instead, we had to consider how creative systems were received by society.

AGD & Computational Creativity

This emphasis on people within Computational Creativity affected the development of ANGELINA in particular.

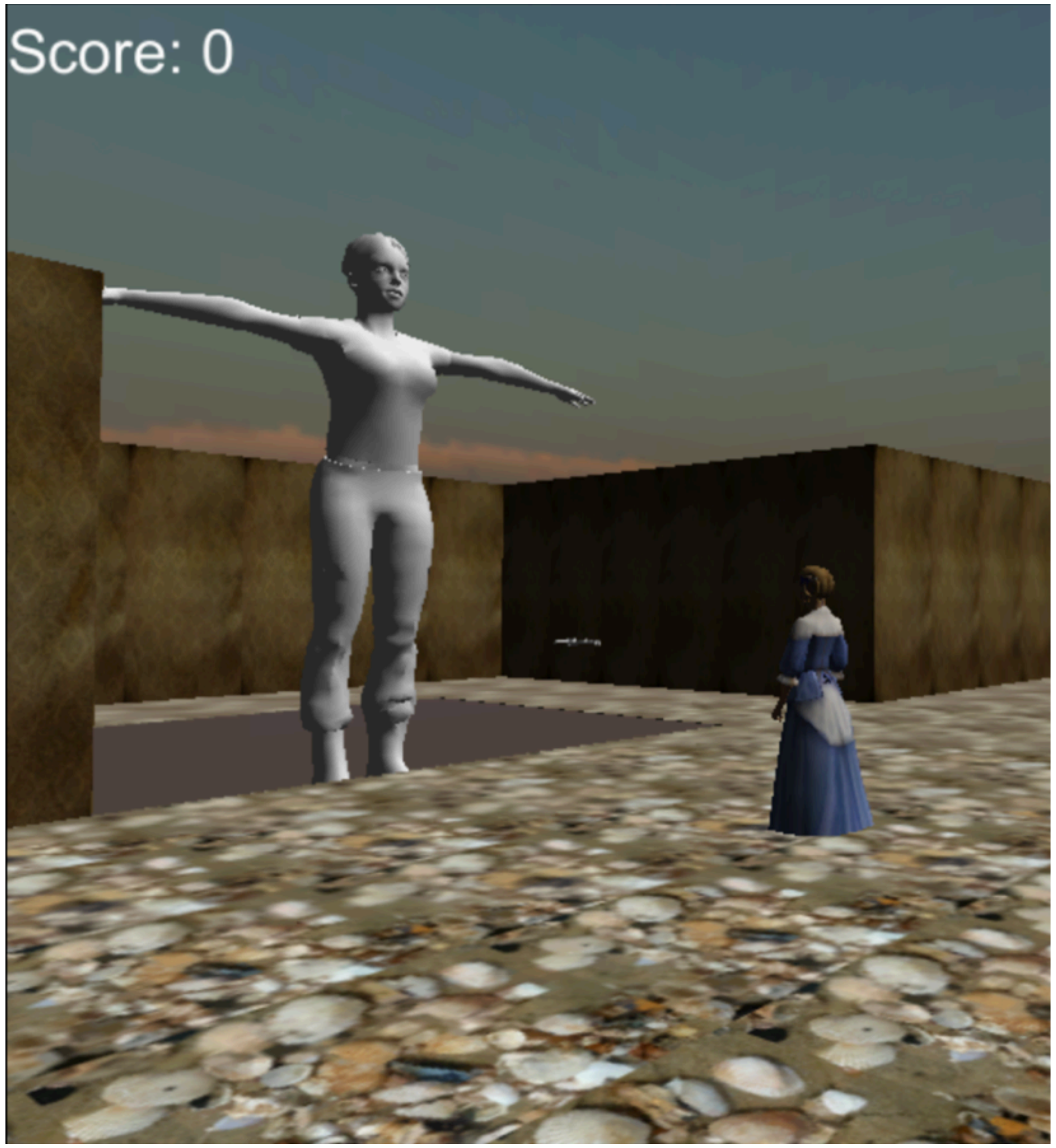
ANGELINA v3 produced **commentaries** describing the decisions it made, to support the audience's interpretations.

> ***I also wanted to include some of the important people from the article. For example, I looked for photographs of James Murdoch. I searched for angry photos of the person because I don't like them.***

The people interested in ANGELINA - developers, journalists, curators, critics, players - were becoming as important, or more important, than the games ANGELINA made.

ANGELINA v5

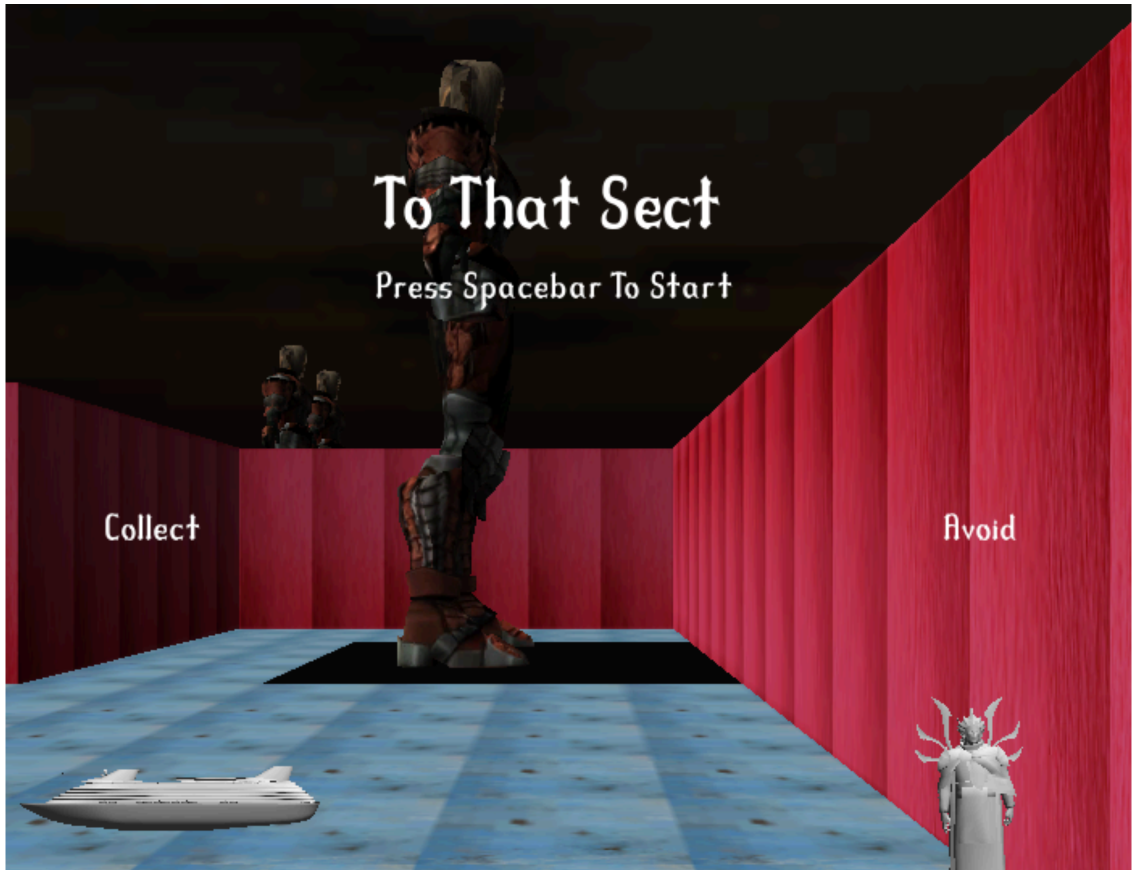
Cook, M, and Colton, S. *Ludus Ex Machina: Building A 3D Game Designer That Competes Alongside Humans* (2014)



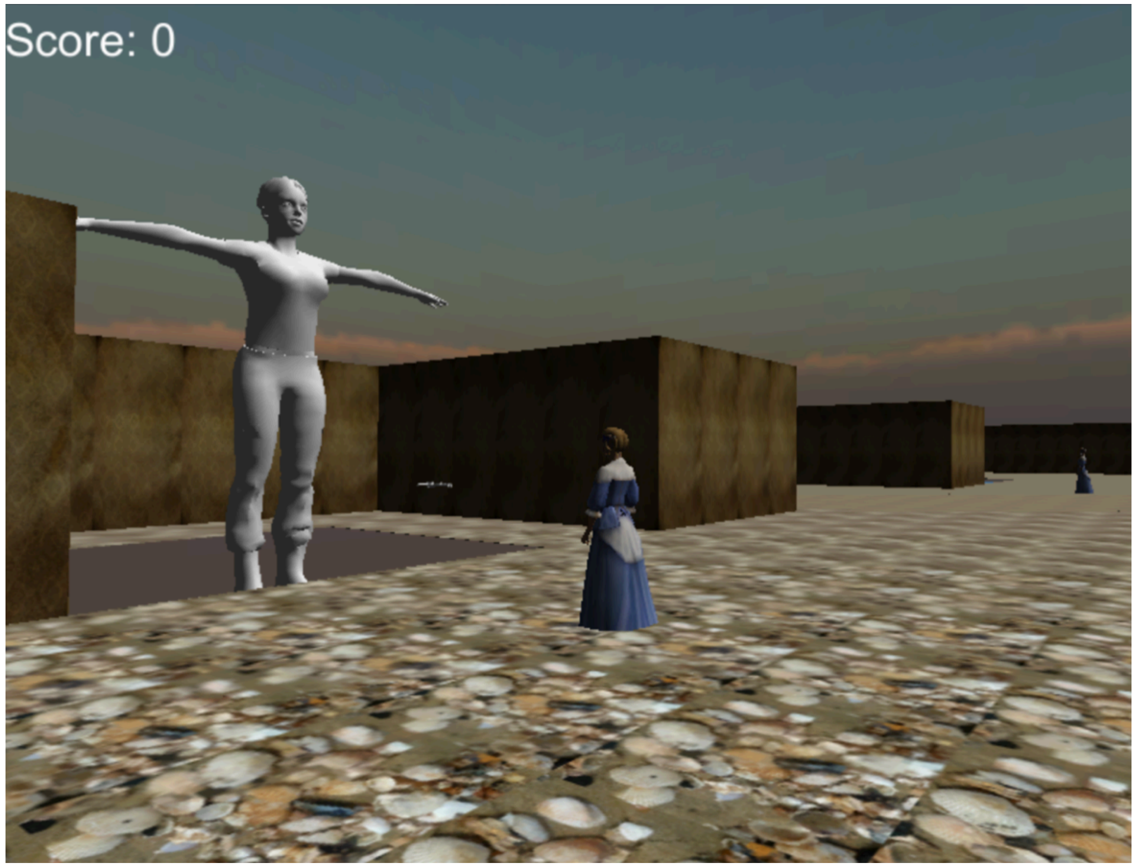
I wanted ANGELINA to become a bigger part of game development communities - to be a game designer, not a game generator.

v5 was a new version of the software that could generate games from any text prompt, with the aim of **entering a game jam.**

Example Games:
To That Sect & Stretch Bouquet Point



To That Sect was ANGELINA's official entry to Ludum Dare

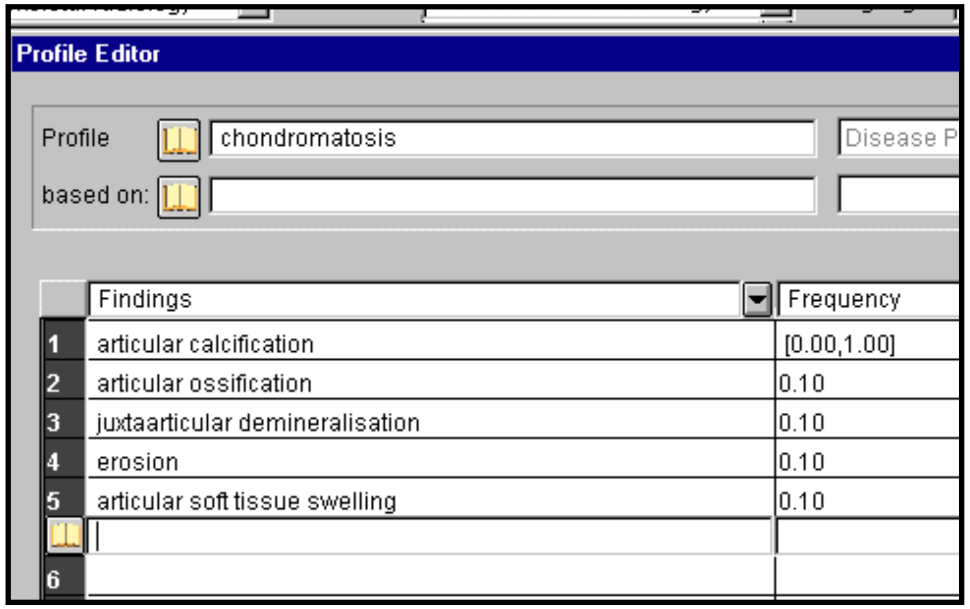


Stretch Bouquet Point was also entered, anonymously.

(We showed that people were *positively* biased towards ANGELINA)

This led to even more exploration of ANGELINA's role in the games community, like exhibiting its work

What is an automated game designer?



Profile Editor

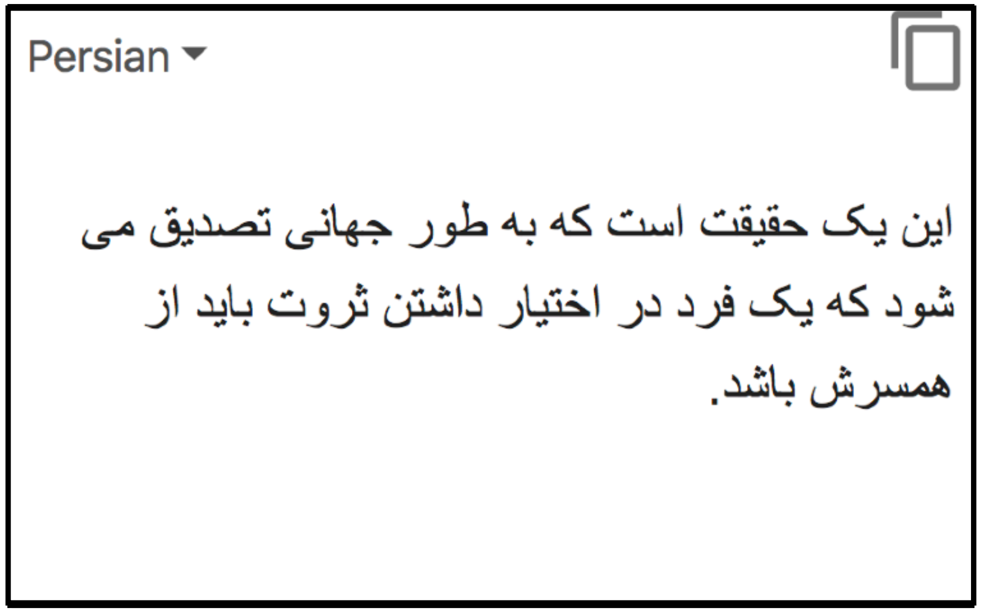
Profile chondromatosis Disease P

based on:

	Findings	Frequency
1	articular calcification	[0.00,1.00]
2	articular ossification	0.10
3	juxtaarticular demineralisation	0.10
4	erosion	0.10
5	articular soft tissue swelling	0.10
6		

Medical Diagnosis

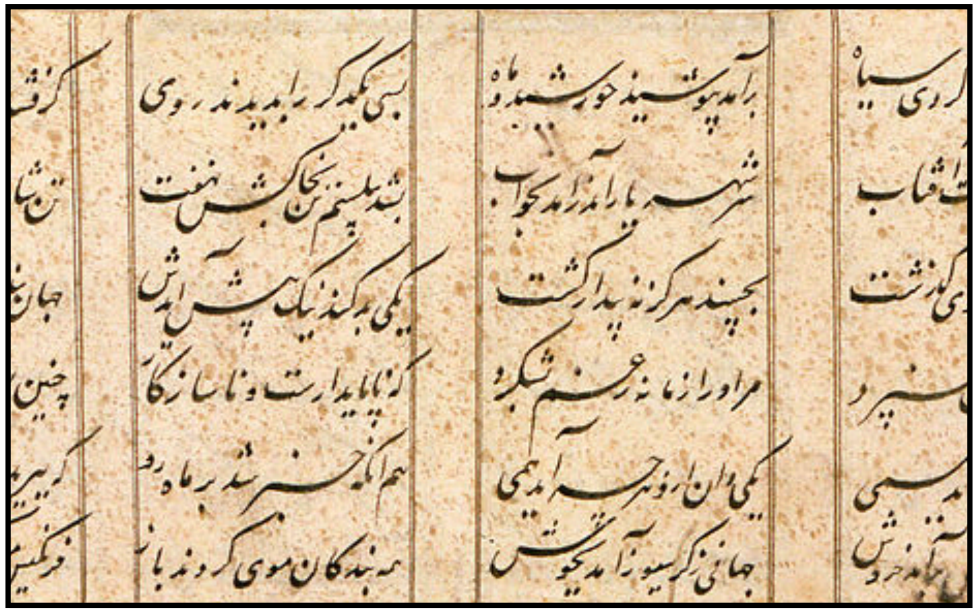
Doctors



Persian

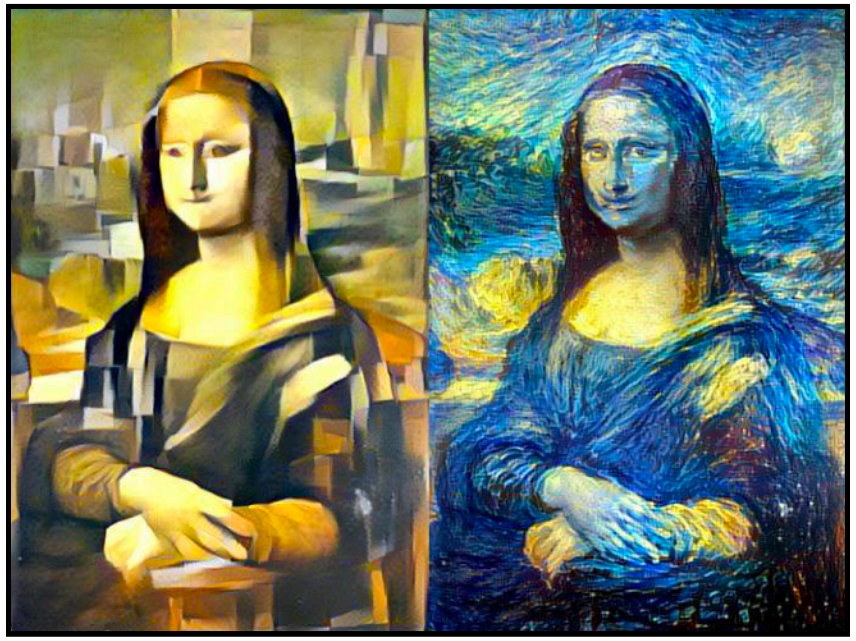
این یک حقیقت است که به طور جهانی تصدیق می
شود که یک فرد در اختیار داشتن ثروت باید از
همسرش باشد.

Machine Translation

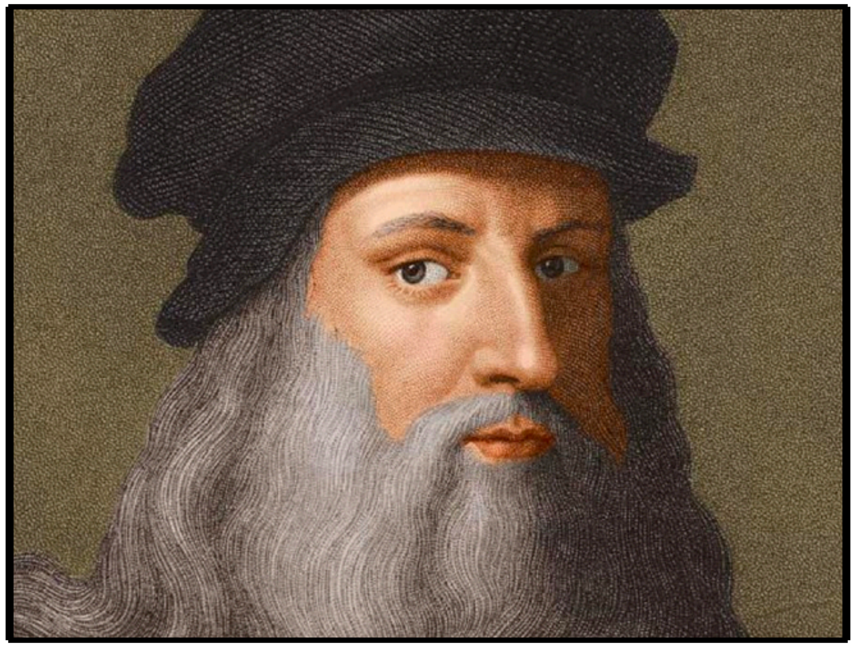


Translators

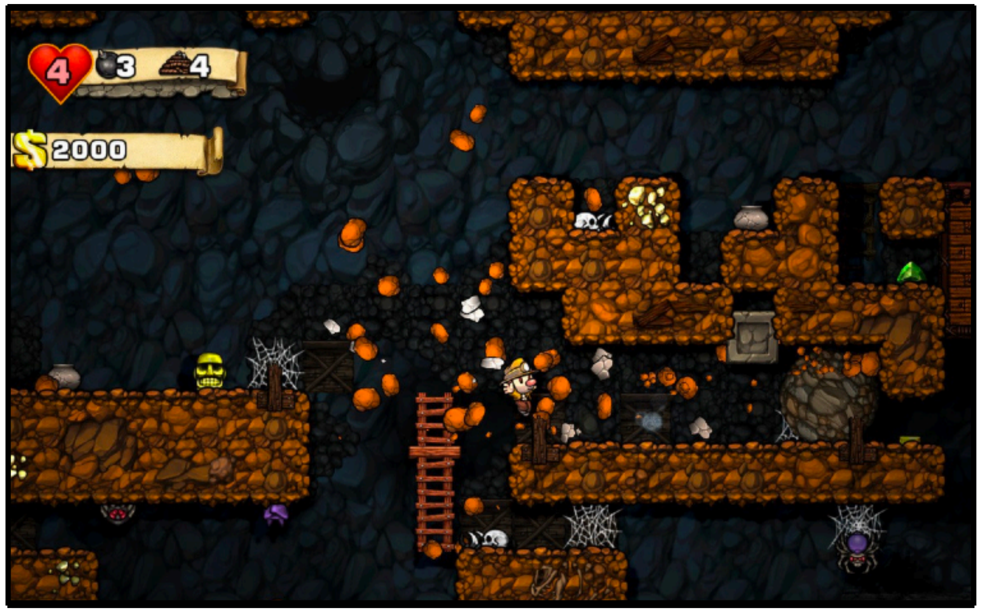
Tasks vs People



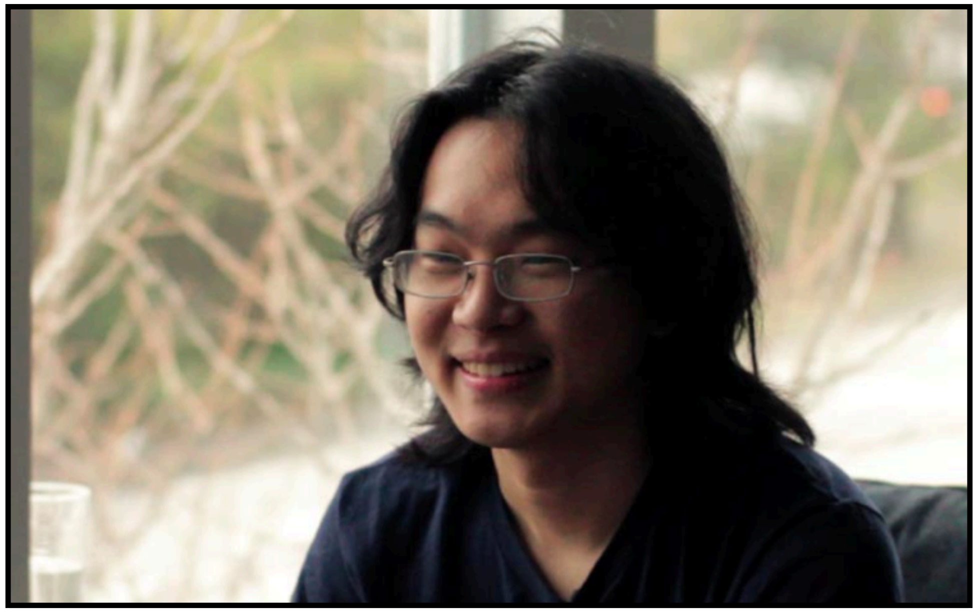
Style Transfer



Artists



Content Generation



Game Developers

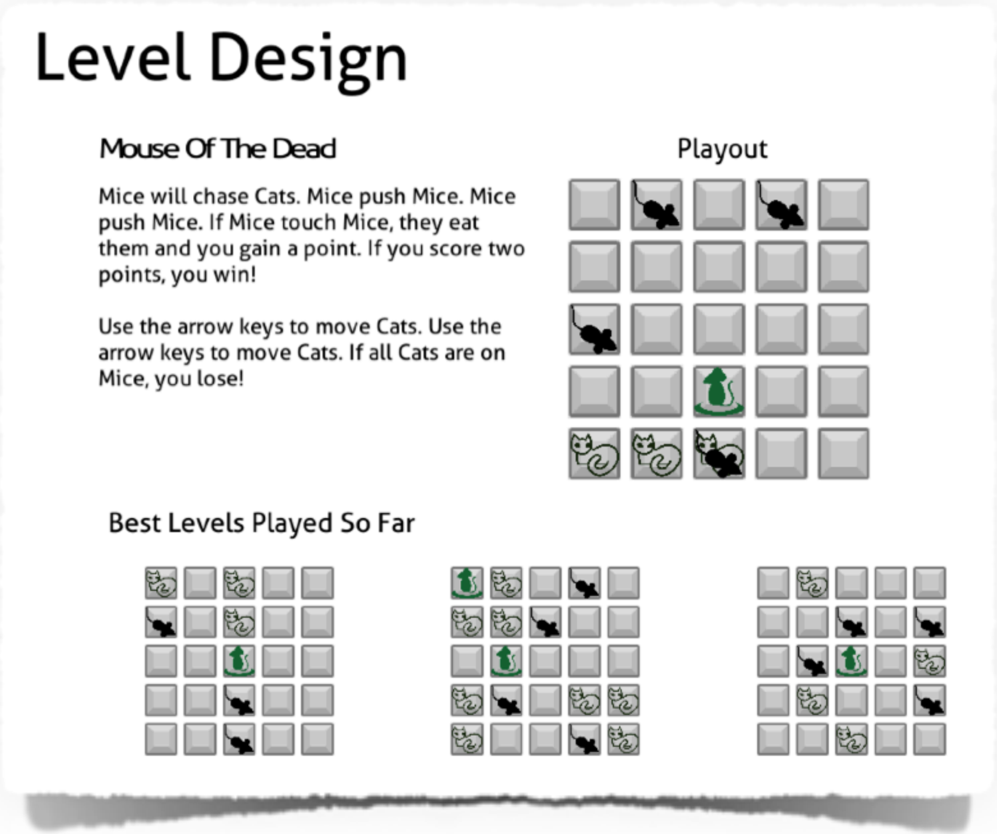
Tasks vs People

Automating *Game Design?*

or

Automating *Game Designers?*

ANGELINA X

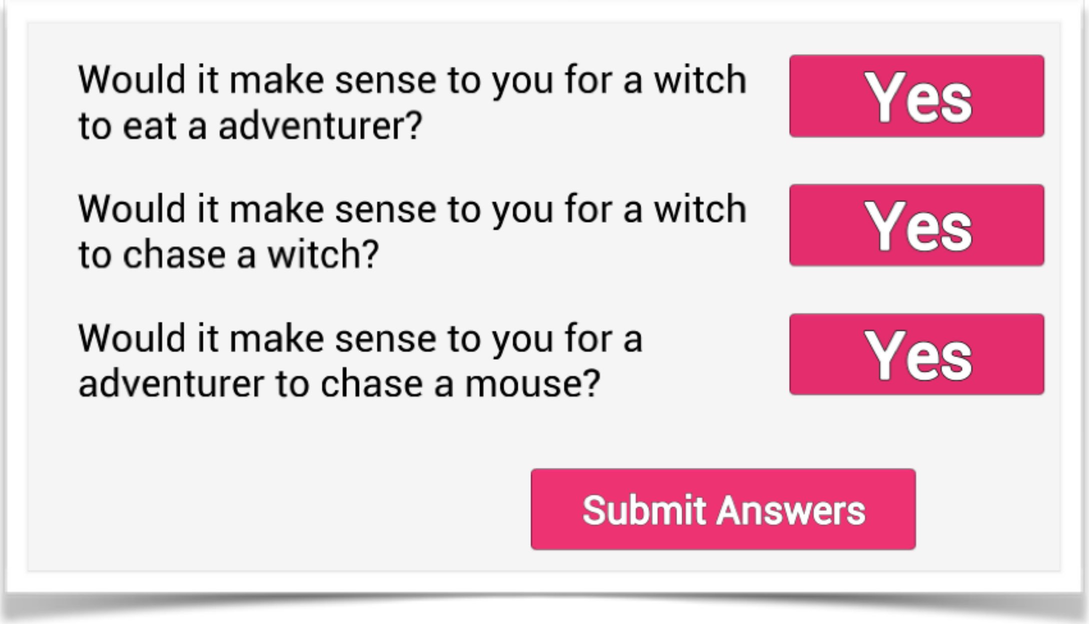


Level Design

Mouse Of The Dead

Mice will chase Cats. Mice push Mice. Mice push Mice. If Mice touch Mice, they eat them and you gain a point. If you score two points, you win!

Use the arrow keys to move Cats. Use the arrow keys to move Cats. If all Cats are on Mice, you lose!



Would it make sense to you for a witch to eat a adventurer?

Would it make sense to you for a witch to chase a witch?

Would it make sense to you for a adventurer to chase a mouse?

Cook, M, and Colton, S. *Redesigning Computationally Creative Systems For Continuous Creation* (2018)

Proposes **continuous creativity** in which an automated game designer does more than just design games.

ANGELINA X is designed with community interaction in mind: every part of the design process can be watched, and there are many vectors for interaction.

Continuous Creation

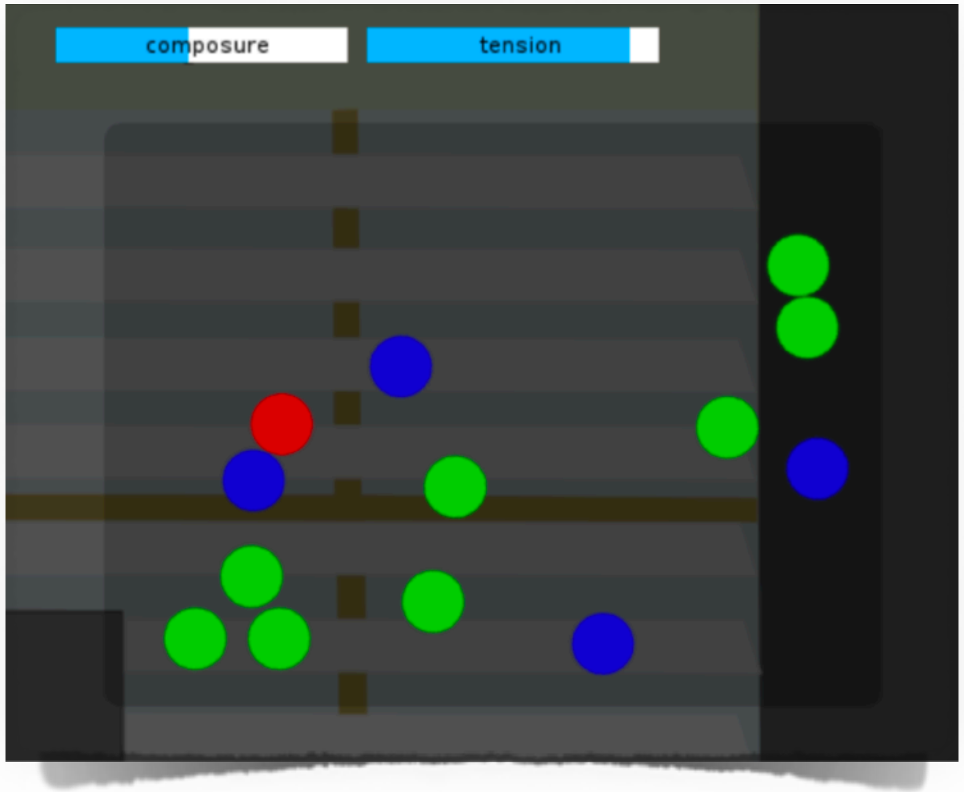
Most AGD systems are switched on, make a game, and then switch off. Continuous creation suggests that games should be a *side effect* of an endless creative process.

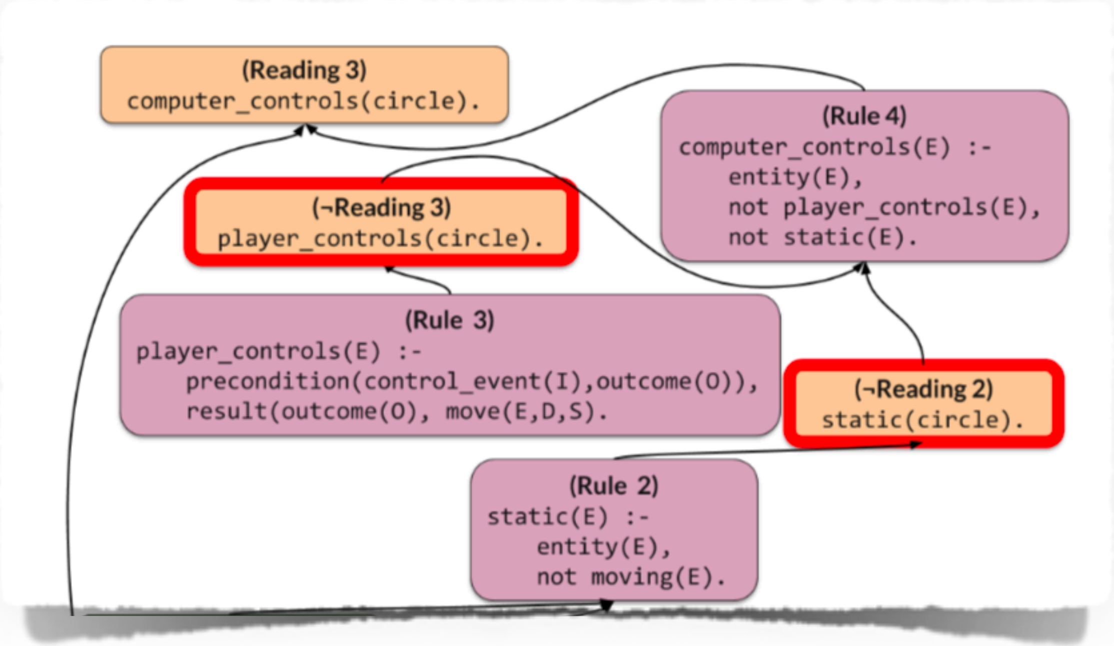
An AGD system should do more than just make games.

- Play games made by other people
- Prototype new ideas
- Playtest its games with people
- Write about its process and progress
- Learn about the world

ANGELINA X is early on in this process, but unlike previous iterations, it’s designed to be slowly extended and updated.

Gemini





(Reading 3)
```
computer_controls(circle).
```

(Rule 4)
```
computer_controls(E) :-
    entity(E),
    not player_controls(E),
    not static(E).
```

(¬Reading 3)
```
player_controls(circle).
```

(Rule 3)
```
player_controls(E) :-
    precondition(control_event(I),outcome(O)),
    result(outcome(O), move(E,D,S).
```

(¬Reading 2)
```
static(circle).
```

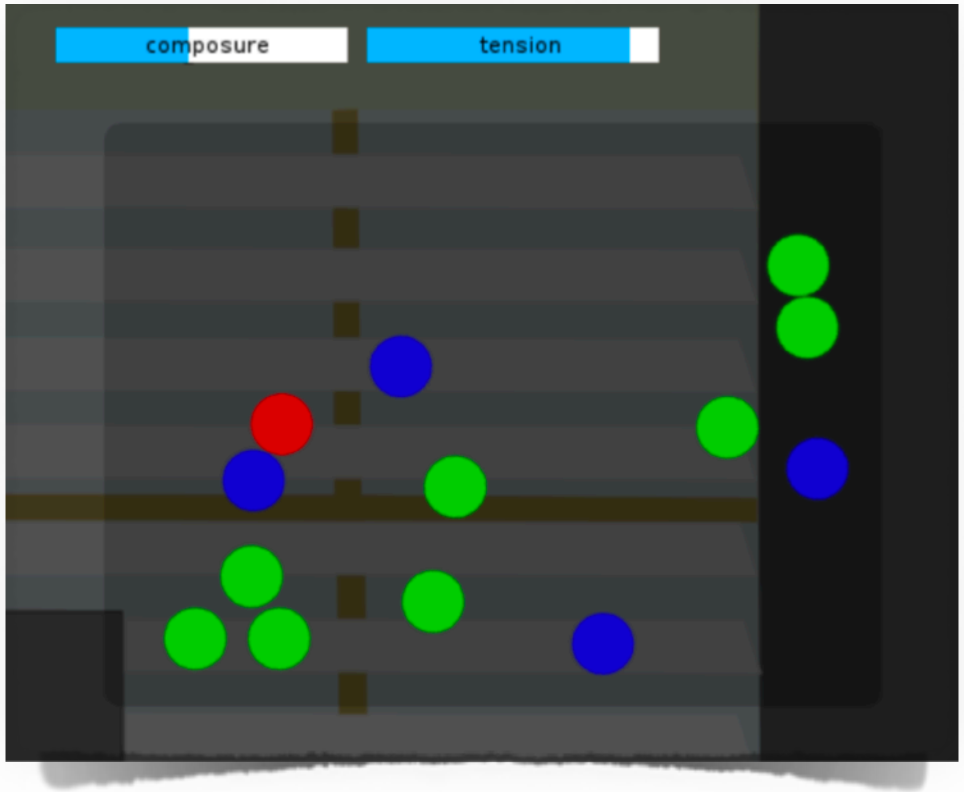
(Rule 2)
```
static(E) :-
    entity(E),
    not moving(E).
```

Summerville, Martens et al. *Gemini: Bidirectional Generation and Analysis of Games via ASP* (2018)

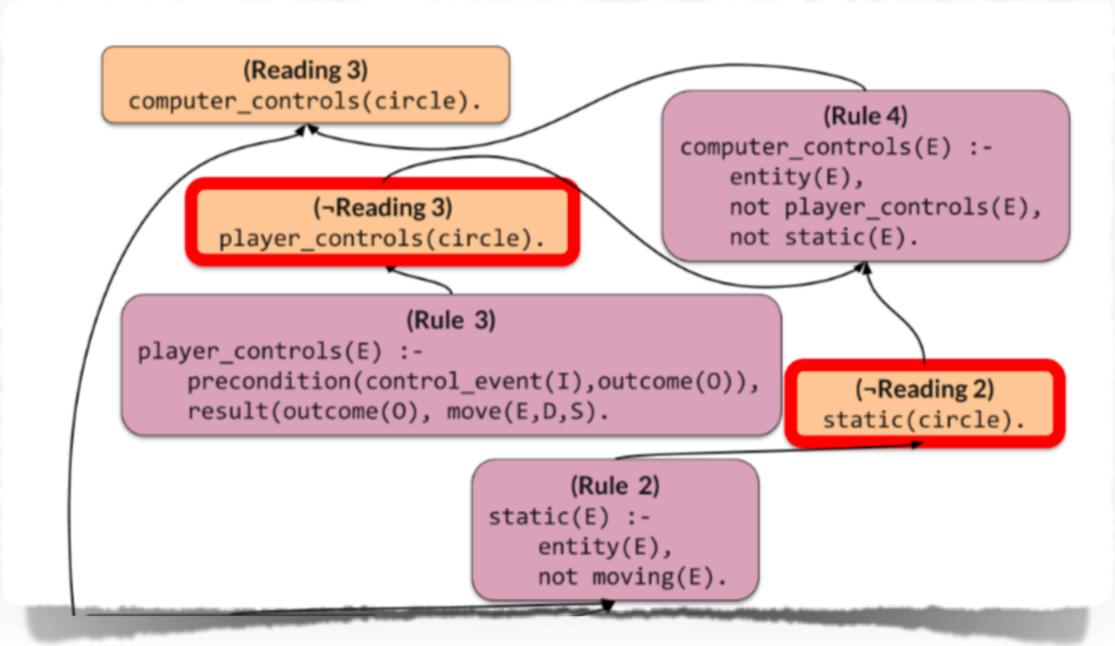
An automated game designer capable of designing games from specifications, and inferring specifications from game designs.

The project originated out of UC Santa Cruz, and develops the ideas of past systems (as well as adding lots of new ideas).

Gemini



One of the most important aspects of Gemini is the emphasis on bidirectional reasoning.



(Reading 3)

```
computer_controls(circle).
```

(Rule 4)

```
computer_controls(E) :-
    entity(E),
    not player_controls(E),
    not static(E).
```

(¬Reading 3)

```
player_controls(circle).
```

(Rule 3)

```
player_controls(E) :-
    precondition(control_event(I),outcome(O)),
    result(outcome(O), move(E,D,S).
```

(¬Reading 2)

```
static(circle).
```

(Rule 2)

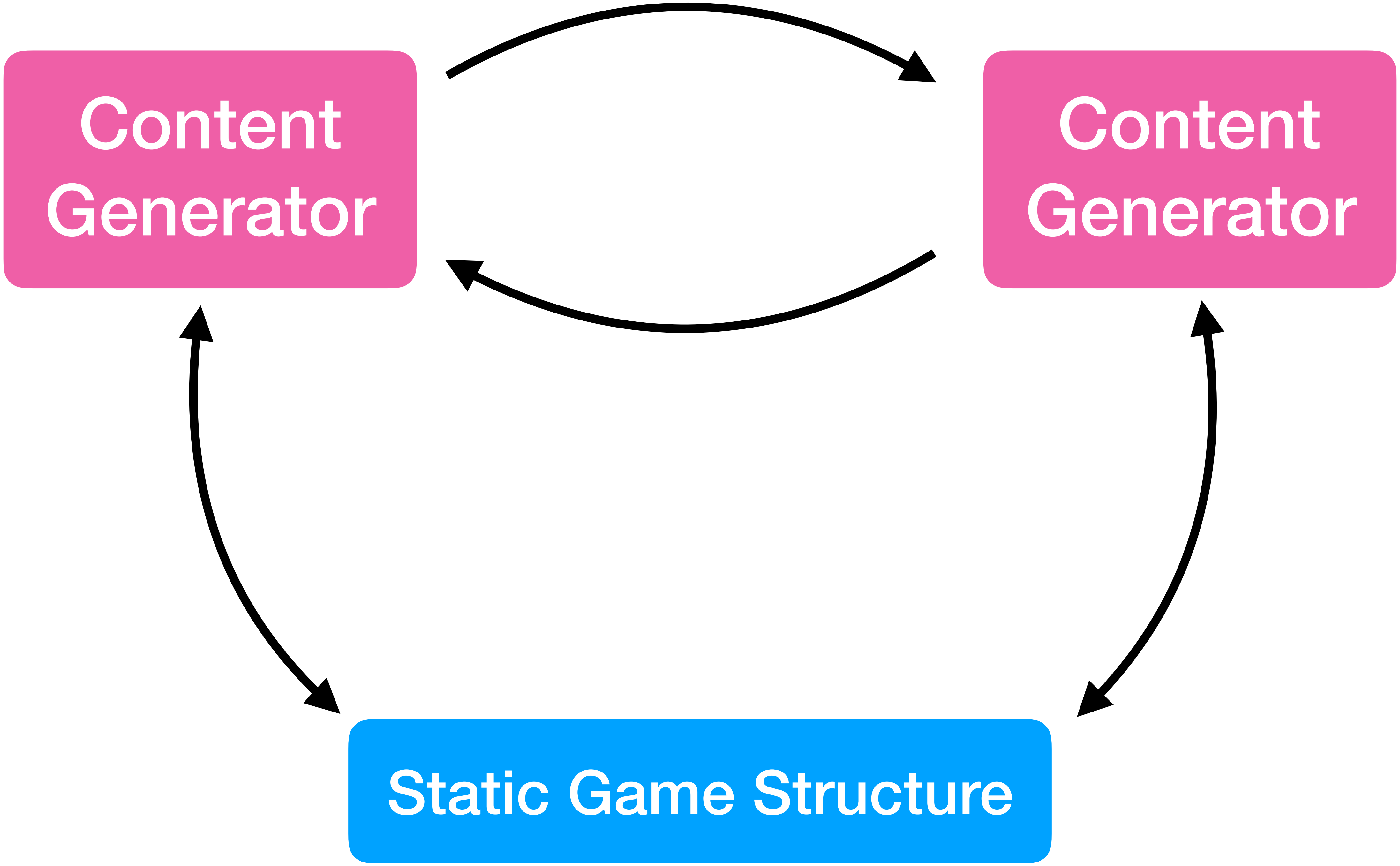
```
static(E) :-
    entity(E),
    not moving(E).
```

Gemini can design games based on maps, but it can also derive readings from games, including ones it did not make.

This is a step towards building systems that can engage in dialogue with other people, and argue for a case.

Automated Game Design

2nd Generation



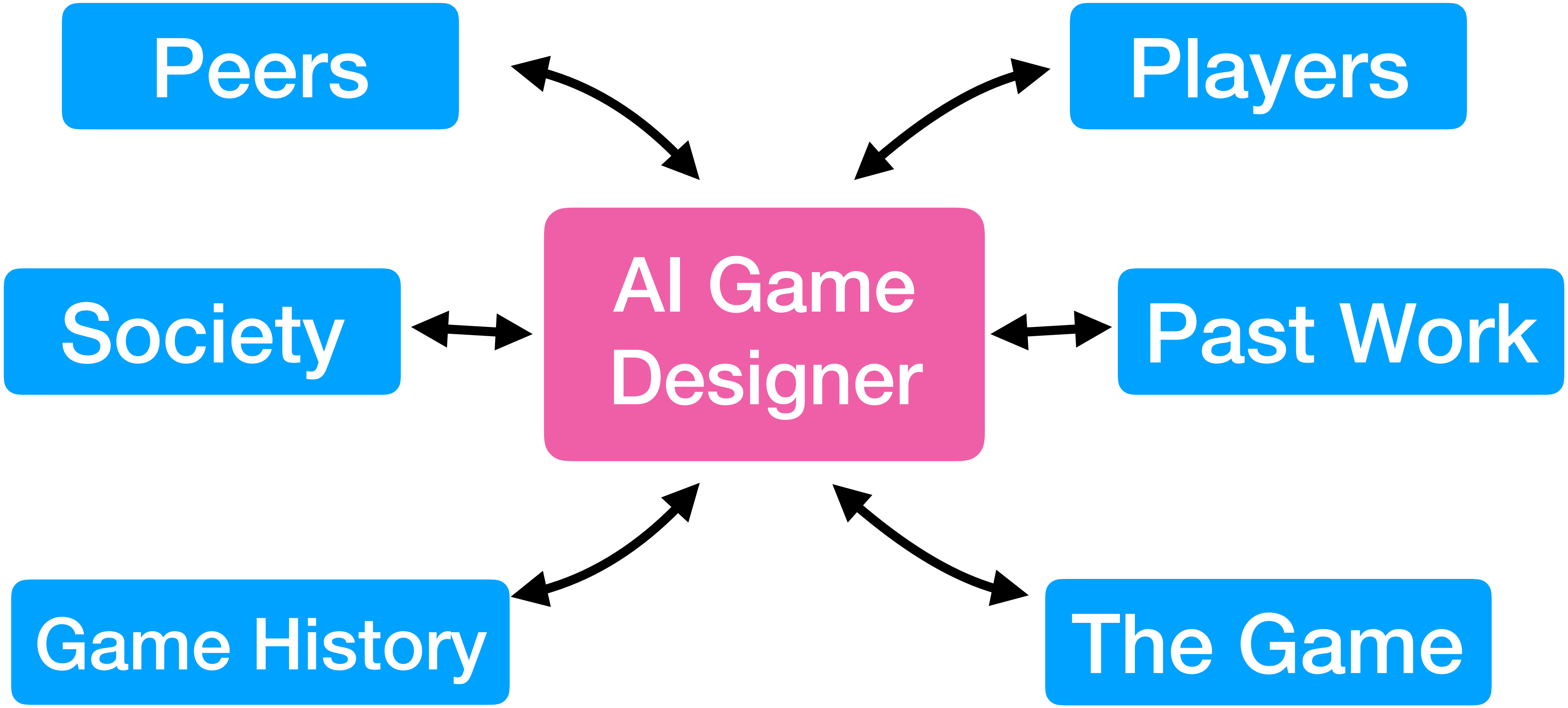
Content Generator

Content Generator

Static Game Structure

Automated Game Design

3rd Generation



Peers

Players

Society

AI Game Designer

Past Work

Game History

The Game

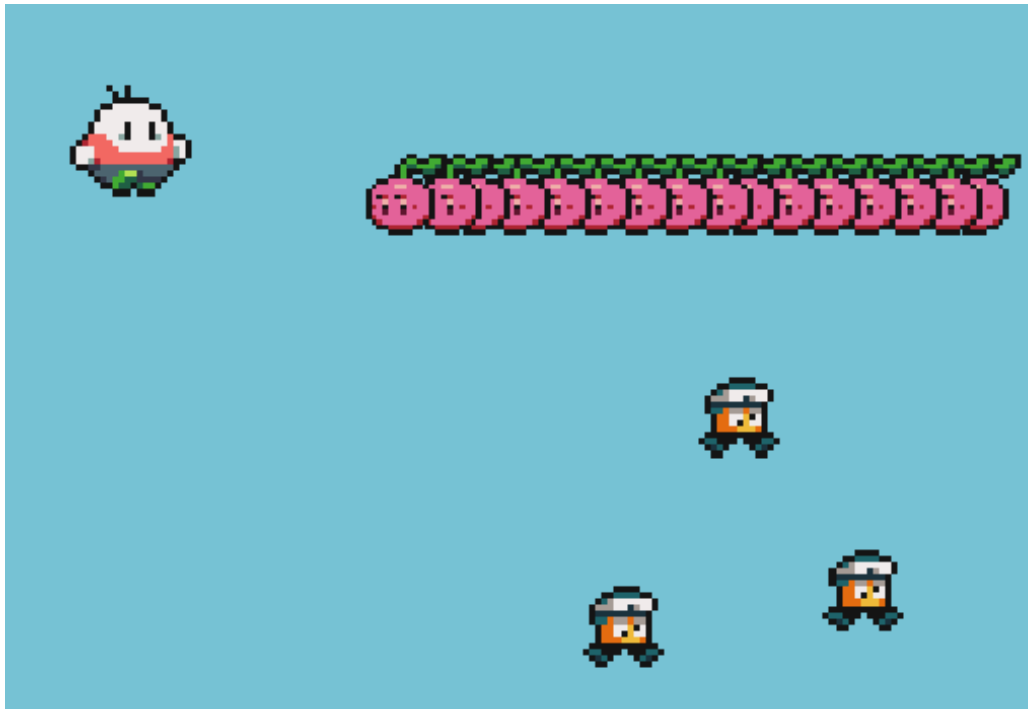
Automated Game Design

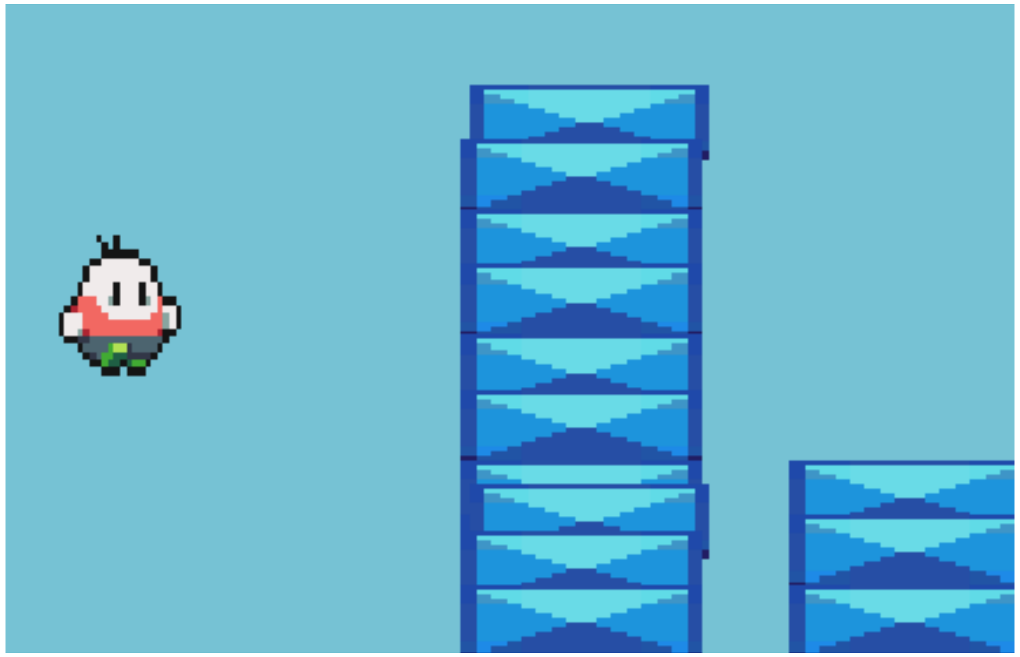
3rd Generation

“Automated game design is ***context-aware*** *procedural content generation”*

Beyond Today & Conclusions

Guzdial & Riedl



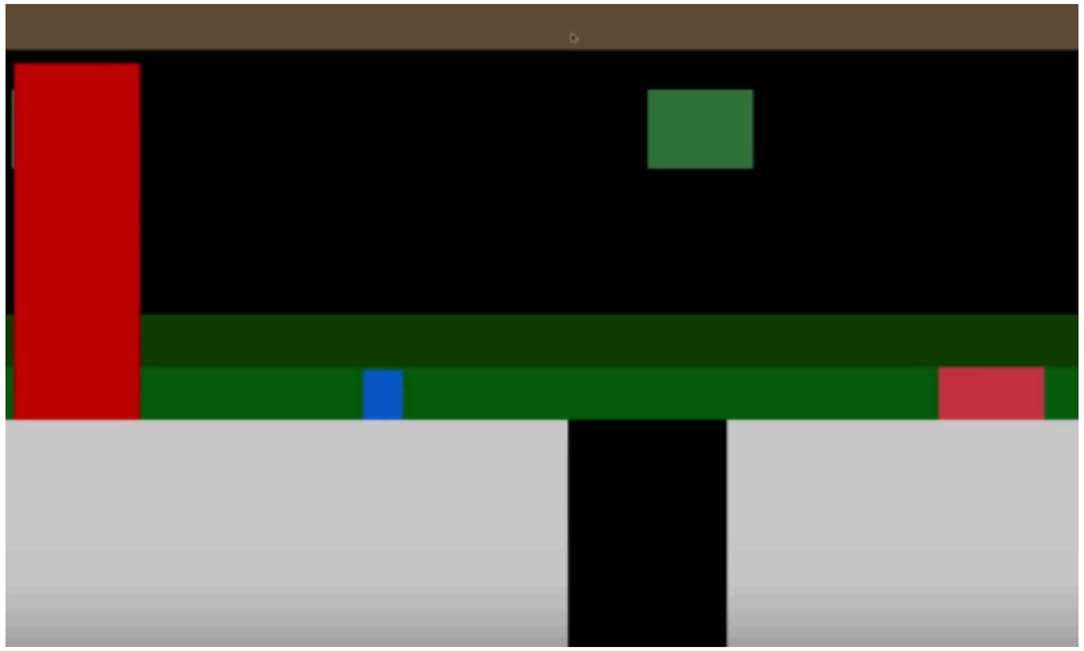


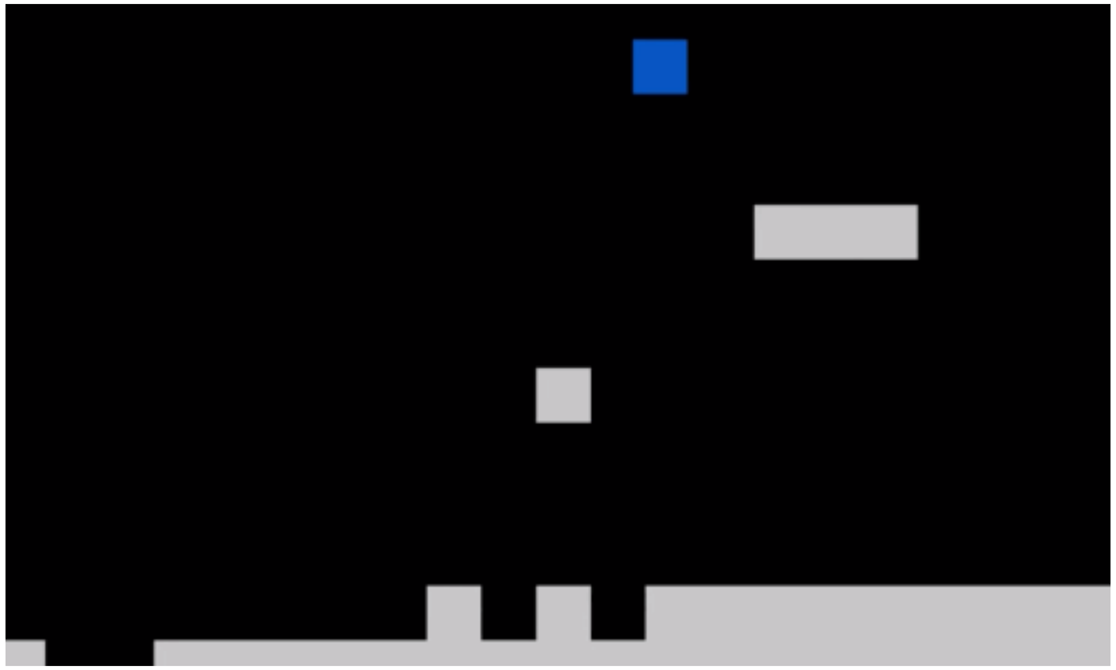
Guzdial, M and Riedl, M.
Automated Game Design via Conceptual Expansion (2018)

Trained a machine learning system on models of existing games, and then used the model to generate new game designs.

Able to produce inventive mechanical blends (such as disappearing bouncy floor tiles) as well as more traditional games.

AGD & Machine Learning





This shows the potential for systems that **learn** from designers, and **innovate** with that knowledge

In some ways this is a return to the early days of AGD - a more opaque system, an emphasis on outcome not process.

But it's also potentially a new way to build deep, rich systems that take into account all kinds of context and nuance.

It's another exciting future for AGD.

Juicy Game Generation

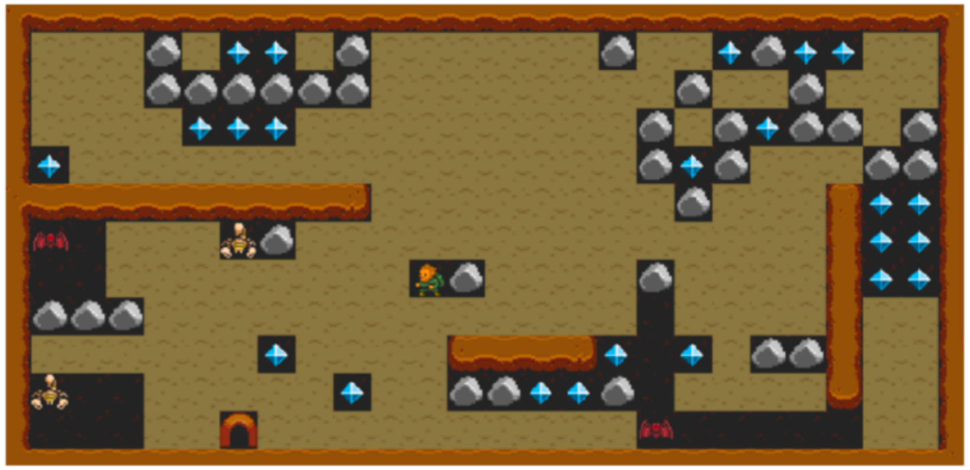
Mads Johanssen is a game designer who started a PhD at ITU Copenhagen in 2018.

He’s interested in studying game description languages and how they differ from languages used by human designers.

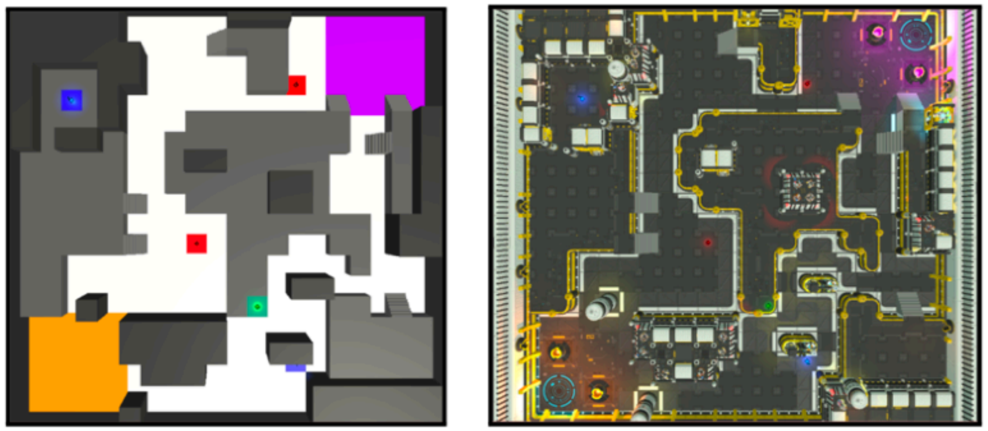
He’s also interested in how AGD systems can work with concepts like **game feel** or **juiciness**.

See “Juice It or Lose It” on YouTube

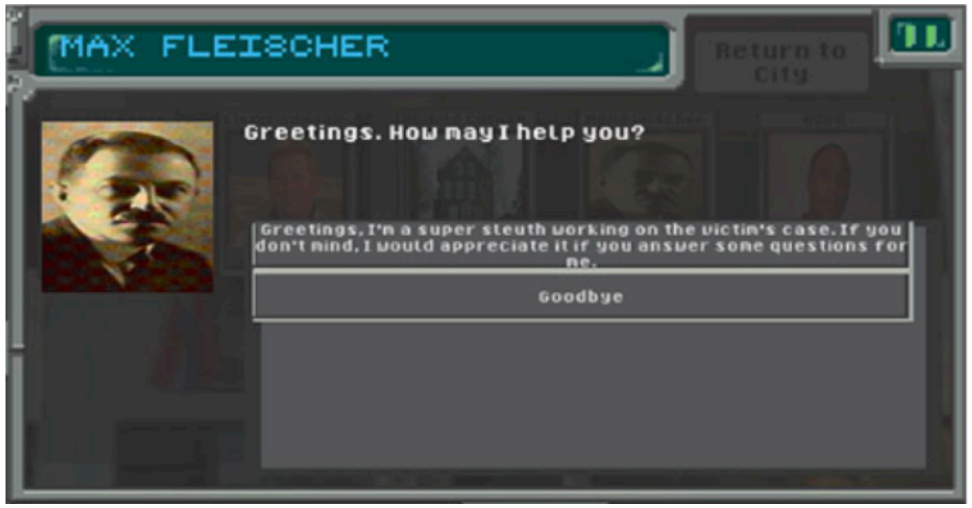
Additional Systems



Khalifa et al. *General Video Game Rule Generation* (2017)



Liapis et al. *Fusing Level and Ruleset Features for Multimodal Learning of Gameplay Outcomes* (2019)

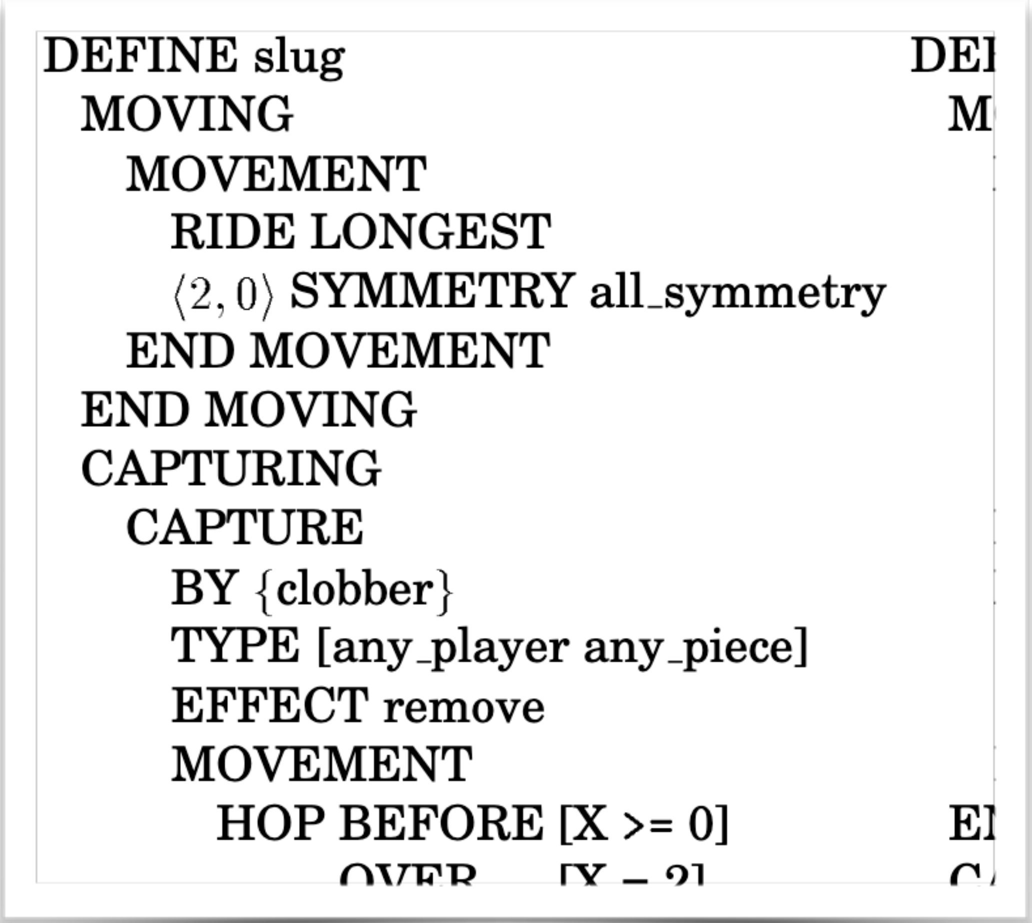


MAX FLEISCHER

Greetings. How may I help you?

Greetings, I'm a super sleuth working on the victim's case. If you don't mind, I would appreciate it if you answer some questions for me.

Barros et al. *Who Killed Albert Einstein? From Open Data to Murder Mystery Games* (2018)



```
DEFINE slug                                   DEI
  MOVING                                       M
    MOVEMENT
      RIDE LONGEST
      ⟨2, 0⟩ SYMMETRY all_symmetry
    END MOVEMENT
  END MOVING
  CAPTURING
    CAPTURE
      BY {clobber}
      TYPE [any_player any_piece]
      EFFECT remove
      MOVEMENT
        HOP BEFORE [X >= 0]                    EI
            OVER      [X = 2]                  C/
```

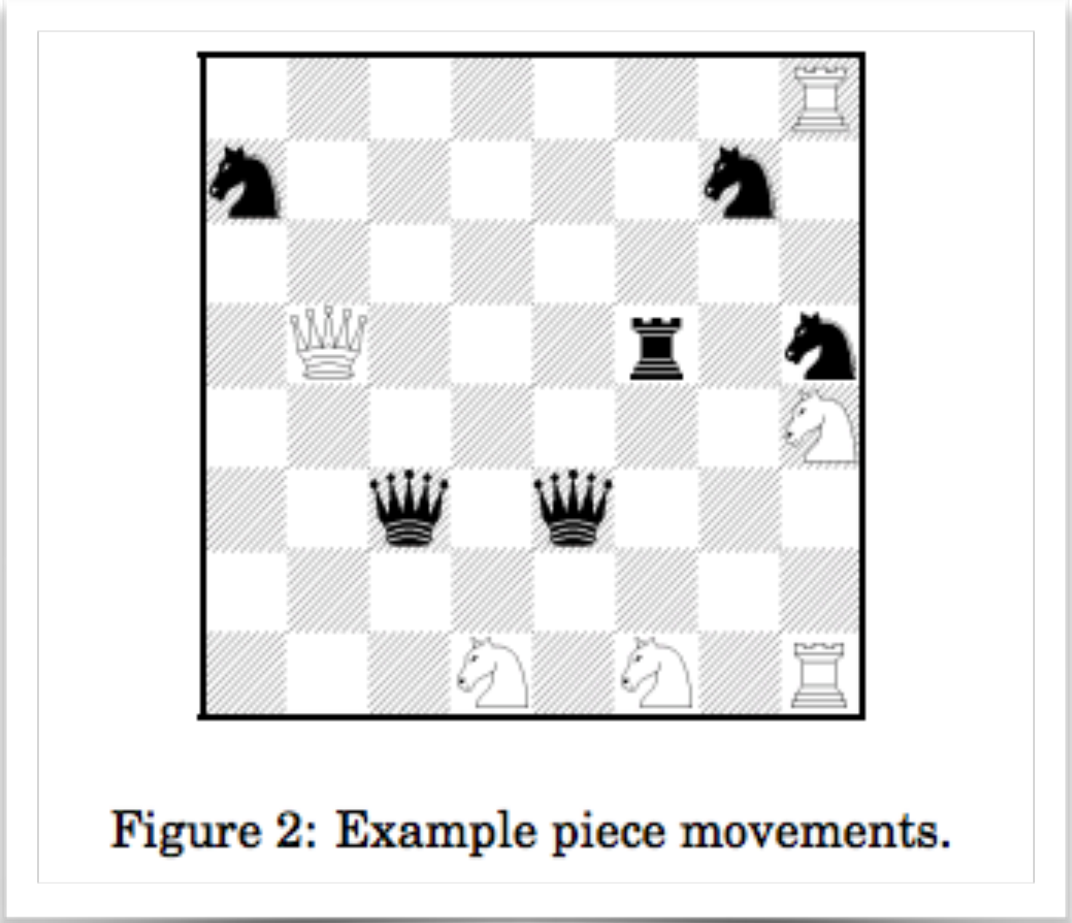
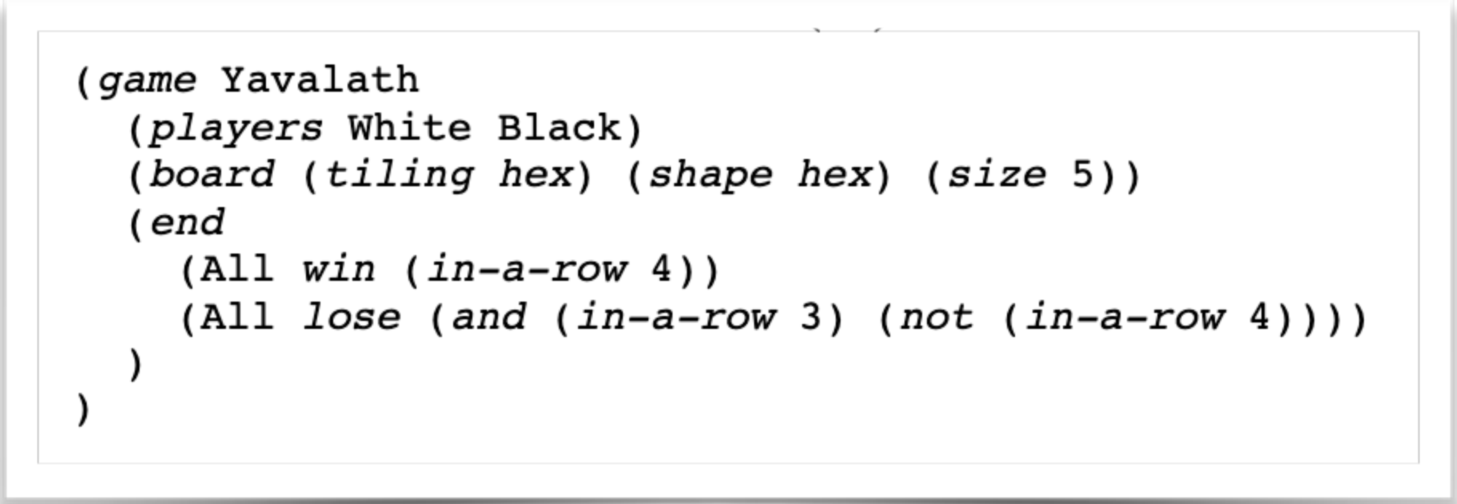


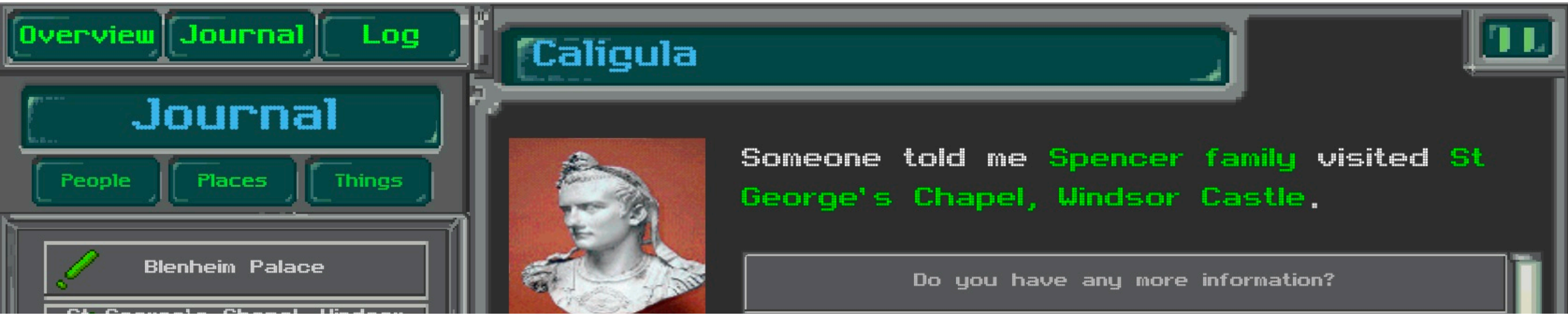
Figure 2: Example piece movements.



```
(game Yavalath
  (players White Black)
  (board (tiling hex) (shape hex) (size 5))
  (end
    (All win (in-a-row 4))
    (All lose (and (in-a-row 3) (not (in-a-row 4))))
  )
)
```

AGD grew out of a view of games-as-rules

Best Levels Played So Far

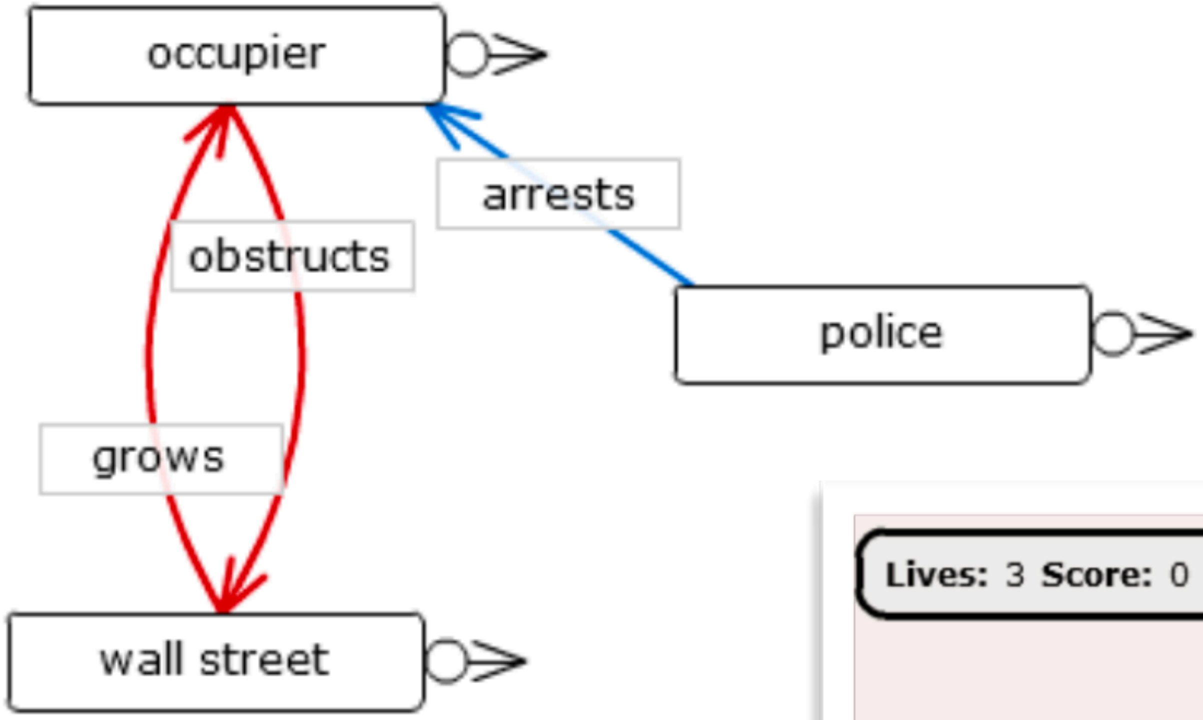


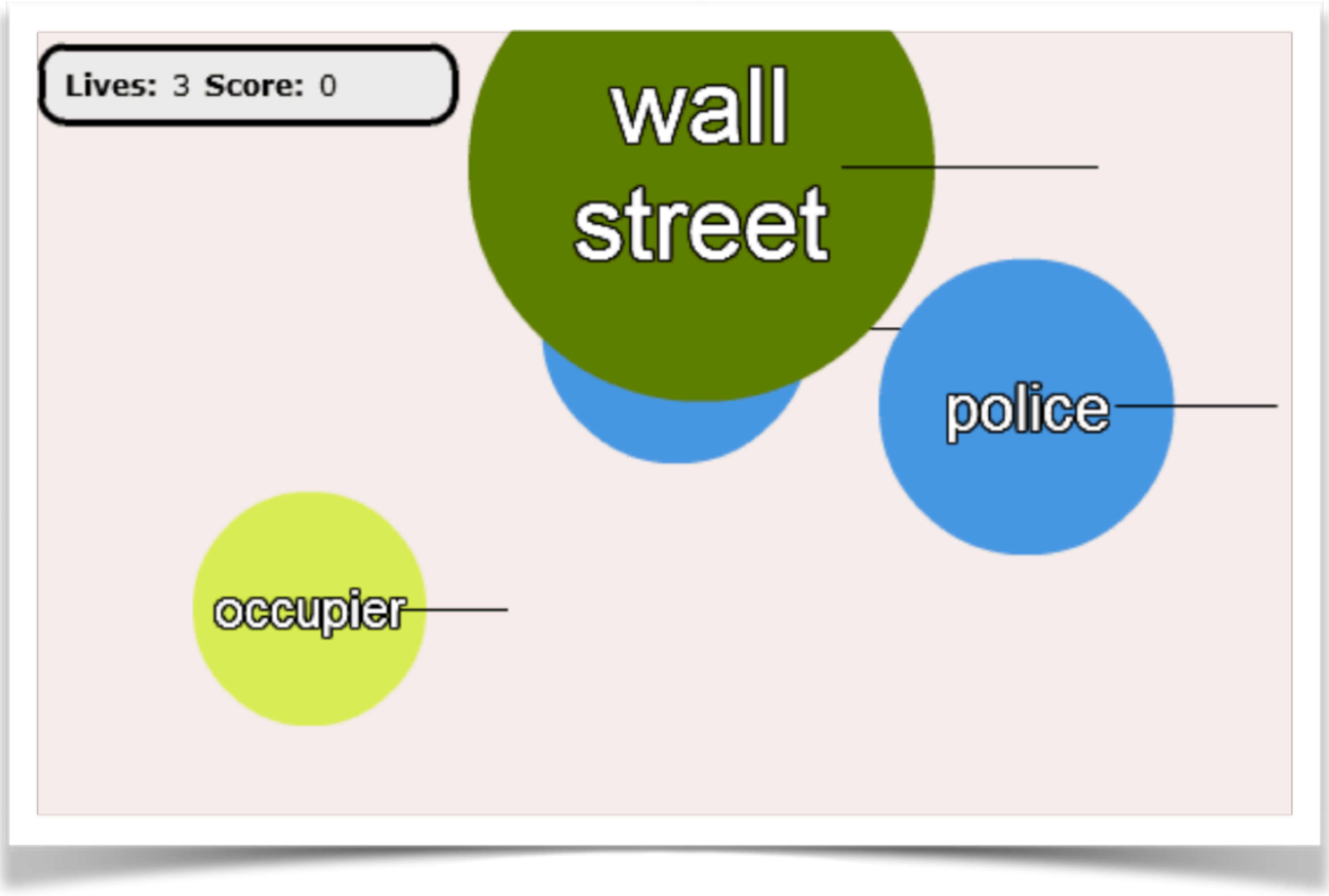
Caligula

Someone told me Spencer family visited St George's Chapel, Windsor Castle.

Do you have any more information?

Today, we have many different perspectives




Lives: 3 Score: 0

No answer is right - but each one encodes a philosophy

Every game designer is different, automated or not.

Automated game design needs more practitioners.

Thanks for listening!

Get in touch:

mike@gamesbyangelina.org or **@mtrc**

Tutorials, papers, blog posts and more from me:

possibilityspace.org

Play lots of ANGELINA's games from all versions:

gamesbyangelina.itch.io

Free tutorials, art and procedural generation resources:

procjam.com