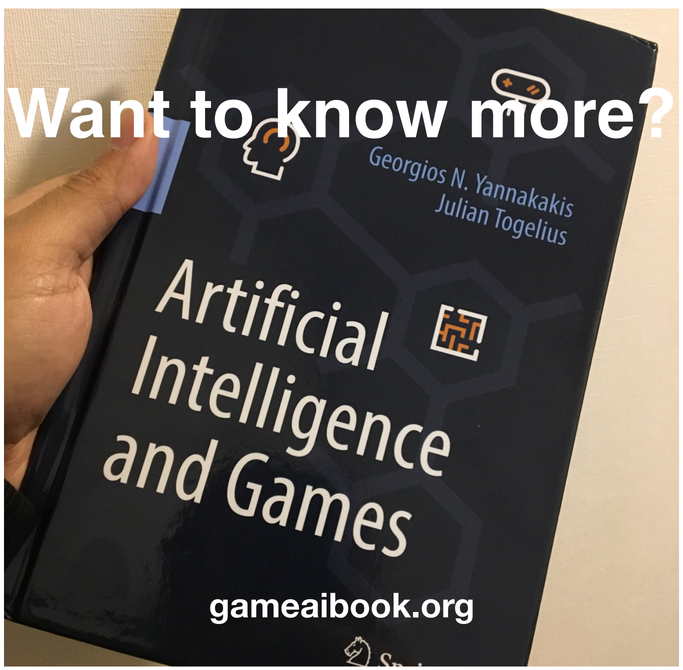
Evolutionary Computation and Games

Julian Togelius, Sebastian Risi,
Georgios N. Yannakakis

NYU
IT University of Copenhagen
L-Università ta' Malta
modl.ai

Who we?



Evolutionary computation can be used to…

- Play games
- Generate game content (levels etc)
- Generate games
- Model players
- Assist designers
- <your idea here>

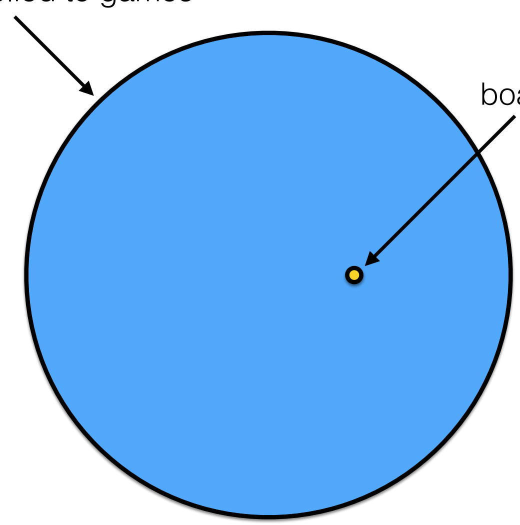
GECCO '20 Companion, July 8–12, 2020, Cancún, Mexico

ACM ISBN 978-1-4503-7127-8/20/07.
https://doi.org/10.1145/3377929.3389854

Playing board games

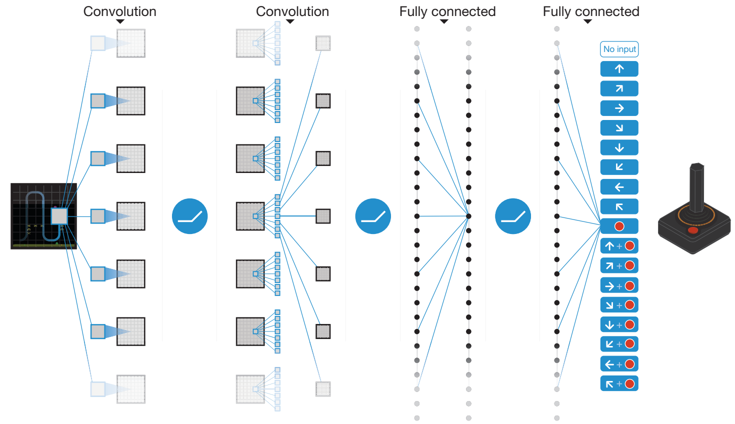
Playing board games

Playing board games



How can evolution be used to play a game?

Common technique: Q-learning with deep nets



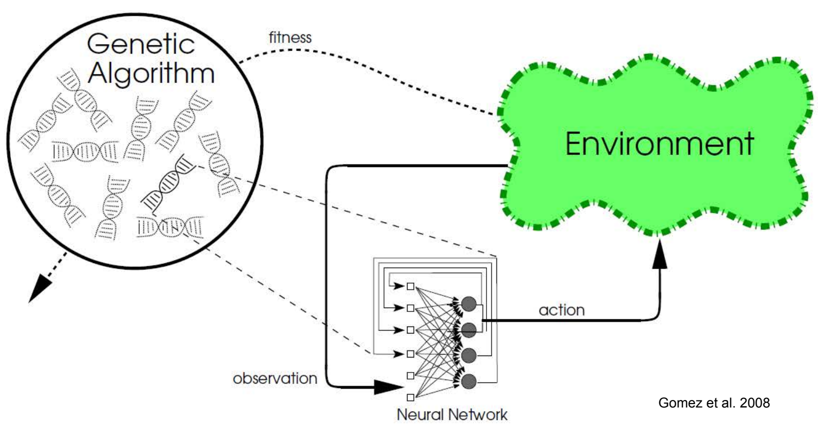
Surely, deep Q-learning is the best algorithm for game-playing!

- Planning (requires forward model)
  - Uninformed search (e.g. minimax, breadth-first)
  - Informed search (e.g. A*)
  - Evolutionary algorithms
- Reinforcement learning (requires training time)
  - TD-learning / approximate dynamic programming
  - Evolutionary algorithms
- Supervised learning (requires play traces to learn from)
  - Neural nets, k-nearest neighbors etc
- Random (requires nothing)

How can evolution be used to play a game?

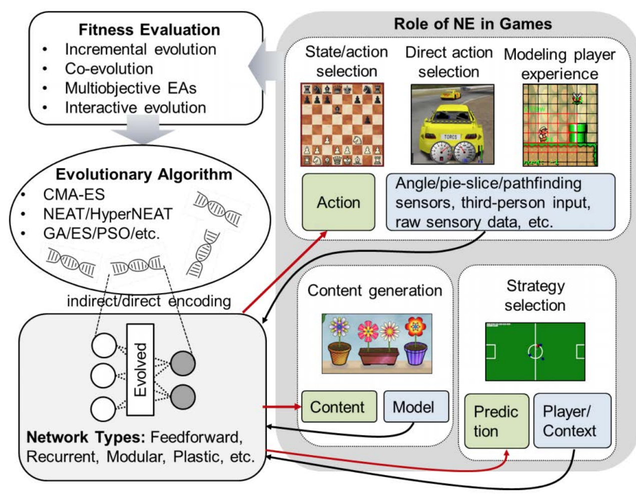
- Evolve an agent that plays the game
  - e.g. through neuroevolution or genetic programming
- Use evolution to play the game (as an action selector)

Neuroevolution



Gomez et al. 2008

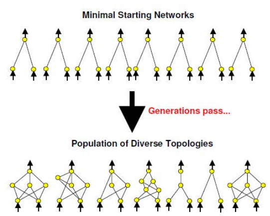
NE Role in Games



Neuroevolution in Games. Risi and Togelius, TCIAIG, 2015.

Evolving Neural Networks

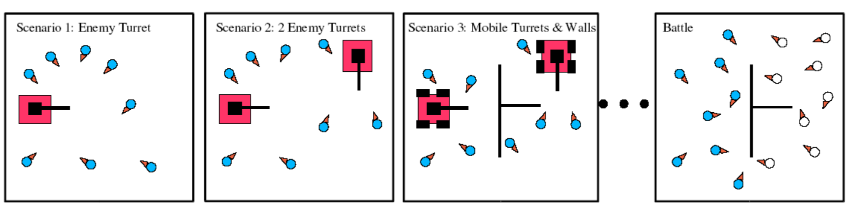
- Direct encodings
  - Evolution strategies / Genetic algorithms
  - NEAT (can evolve arbitrary topologies)
  - Many more …
- Indirect encodings
  - HyperNEAT
  - Compressed weight space
  - Many more …



Why Neuroevolution

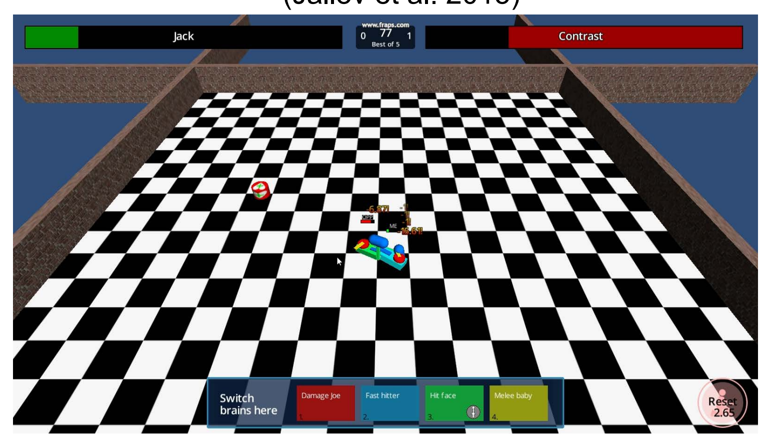
- Broad applicability
- Can be used for both supervised and RL problems
- Diversity
- Open-ended learning
- Enables new types of games

NERO: NeuroEvolving Robotic Operatives (Stanley et al. 2005)



- NPCs improve in real time as game is played
- Player can train AI for goal and style of play
- Each AI Unit Has Unique NN
- Supports incremental evolution

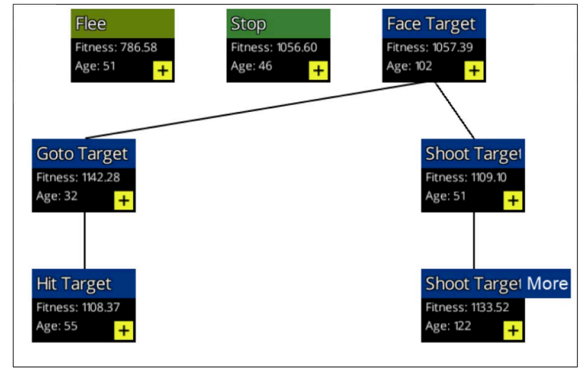
EvoCommander
New game mechanics based on brain switching (Jallov et al. 2015)



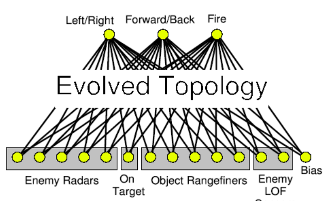
https://www.youtube.com/watch?v=xFwjbCe5Zo8#t=22

Fitness Evaluations in Games

- Co-evolution
- Multiobjective Evolution
- Incremental Evolution



NE Role: Direct action selection

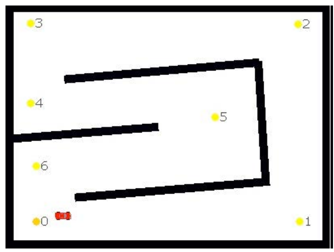


Car racing

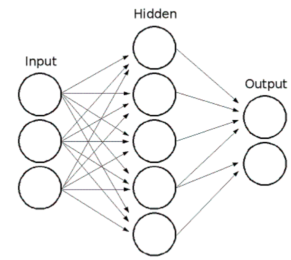
- Driving a car fast requires fine motor control (in both senses)
- Optimizing lap times requires planning
- Overtaking requires adversarial planning

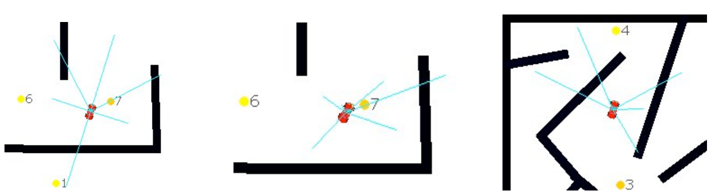
A simple car game

- Walls are solid
- Waypoints must be passed in order
- Fitness: continuous approximation of waypoints passed in 700 time steps



- Inputs
  - Six range-finder sensors (evolvable pos.)
  - Waypoint sensor, Speed, Bias
- Networks
  - Standard multi-layer perceptron, 9:6:2
  - Outputs interpreted as thrust/steering





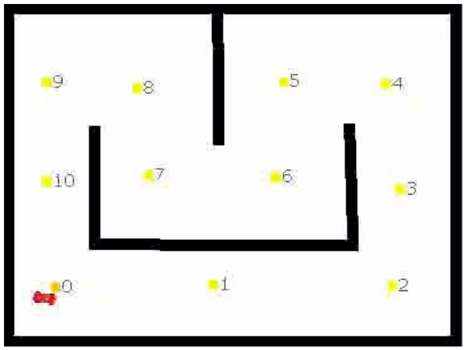
Algorithm 2: Evolution Strategy(μ,λ,n)

```
INITIALIZE(Population, μ + λ individuals)
for i=1 to n do
    for j=1 to (μ + λ) do
        EVALUATE(Population[j])
    end
    PERMUTE(Population)
    SORTONFITNESS(Population)
    for j=μ to (μ + λ) do
        Population[j] ← COPY(Population[j-λ])
        WEIGHTMUTATE(Population[j])
    end
end
```

Mutation: add Gaussian noise with sd 1 to each connection

Fitness: progress around the track

Example video



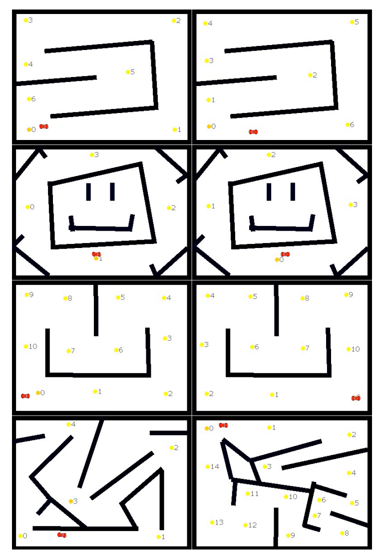
Evolved with 50+50 ES, 100 Generations

Choose your inputs (+their representation)

- Using third-person inputs (cartesian inputs) seems not to work
- Either range-finders or waypoint sensor can be taken away, but some fitness lost
- A little bit of noise is not a problem, actually it's desirable
- Adding extra inputs (while keeping core inputs) can reduce evolvability drastically!

Generalization and specialization

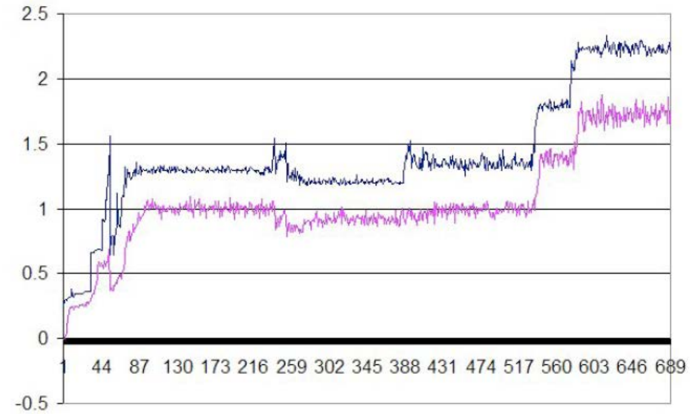
- A controller evolved for one track does not necessarily perform well on other tracks
- How do we achieve more general game-playing skills?
    - Is there a tradeoff between generality and performance?



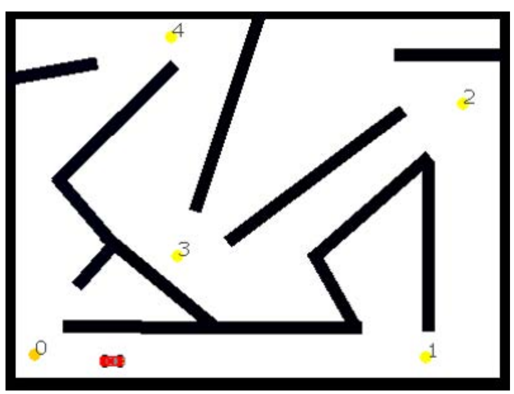
Incremental evolution

- Introduced by Gomez & Mikkulainen (1997)
- Change the fitness function *f* (to make it more demanding) as soon as a certain fitness is achieved
- In this case, add new tracks to *f* as soon as the controller can drive 1.5 rounds on all tracks currently in *f*

Incremental evolution



Video: navigating a complex track



Observations

- Controllers evolved for specific tracks perform poorly on other tracks
- General controllers, that can drive almost any track, can be incrementally evolved
- Starting from a general controller, a controller can be further evolved for *specialization* on a particular track
    - drive faster than the general controller
    - works even when evolution from scratch did not work!

Two cars on a track

- Two car with solo-evolved controllers on one track: disaster
  - they don't even see each other!
- How do we train controllers that take other drivers into account? (avoiding collisions or using them to their advantage)
- Solution: car sensors (rangefinders, like the wall sensors) and *competitive coevolution*

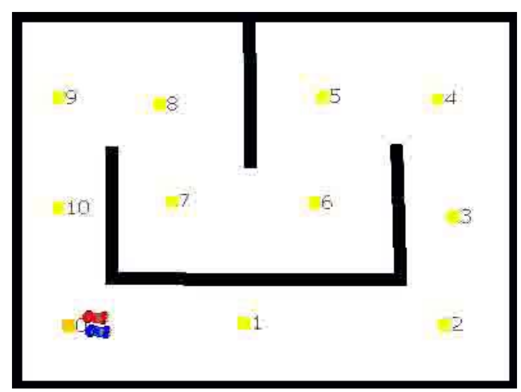
Competitive coevolution

- The fitness function evaluates at least two individuals
- One individual's success is *adversely* affected by the other's (directly or indirectly)
- Very potent, but seldom straightforward; e.g. Hillis (1991), Rosin and Belew (1996)

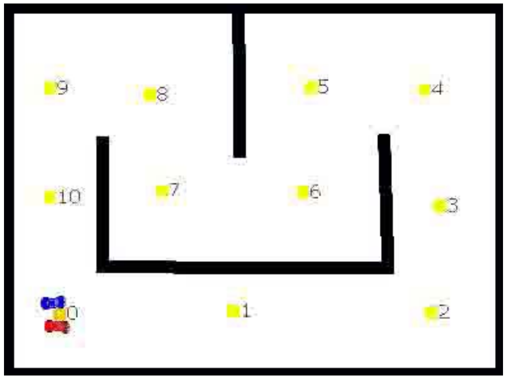
Competitive coevolution

- Standard 15+15 ES; each individual is evaluated through testing against the current best individual in the population
- Fitness function a mix of...
  - Absolute fitness: progress in n time steps
  - Relative fitness: distance ahead of or behind the other car after n time steps

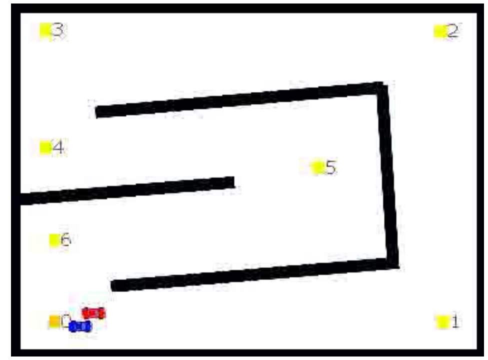
Video: absolute fitness



Video: 50/50 fitness



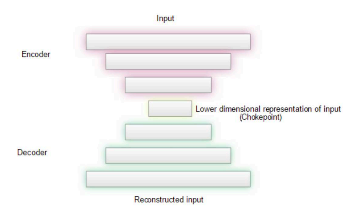
Video: relative fitness

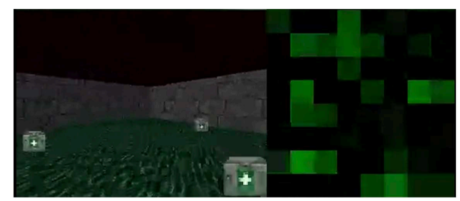


Open Challenges: NE in Games

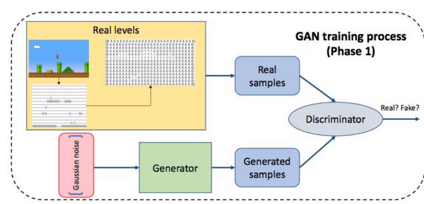
- ~~Reaching Record-beating Performance~~
- Combining evolution with other learning methods
- ~~Learning from high-dimensional/raw data~~
- General video game playing
- Combining NE with life-long learning
- Competitive and cooperative coevolution
- Fast and reliable methods for commercial games

Emerging Trends – Hybrid Methods

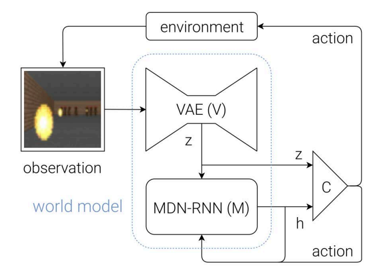




Alvernaz and Togelius, 2017



Volz et al. 2018



Ha and Schmidhuber, 2018

Using evolution to plan?

- Some games have extremely high branching factor
  - Chess: 35
  - Go: 350
  - Civilization/StarCraft: say you have ten units, which can each take one of ten actions…
- Tree search cannot even get past the first ply
- One solution: treat the whole plan as a sequence of actions, the value of the final state as fitness…

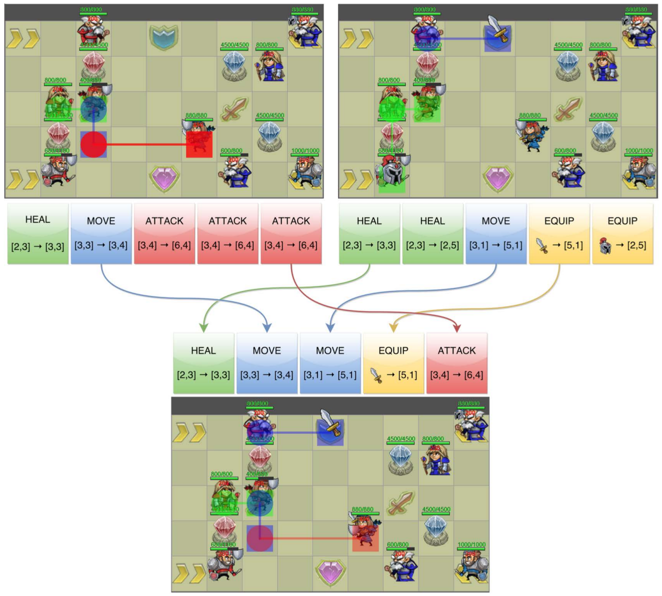
Hero Academy

Enormous branching factor beats MCTS

	Random	Greedy Action	Greedy Turn	MCTS
Greedy Action	100%			
Greedy Turn	100%	64.0%		
MCTS	100%	48.5%	22.0%	

Online Evolutionary Planning

- Evolve the set of actions to take each turn
  - Chromosome is a sequence of five actions
- Simple evolutionary algorithm:
  - Population size of 100, 50% elitism, random selection of parents, uniform crossover, 10% mutation rate



Results: wow

	Random	Greedy Action	Greedy Turn	MCTS
Online Evolution	100%	90.0%	80.5%	98%

- ~10,000 unique outcomes evaluated each turn (6 seconds)
- ~3,500 generations each turn on average

Niels Justesen, Tobias Mahlmann, Sebastian Risi and Julian Togelius (2017): Playing Multi-Action Adversarial Games: Online Evolutionary Planning versus Tree Search. IEEE TCIAIG.

Procedural content generation in games

Why generate game content?

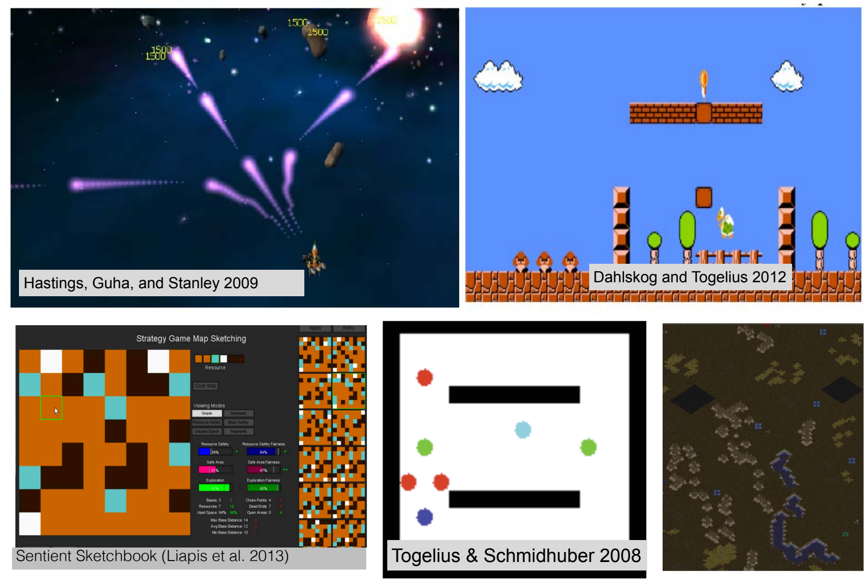
- To replace the human? (Saving time and money…)
- To assist the human designer?
- To make new types of games possible?
- To go beyond human creativity
- To really understand design

Search-based PCG

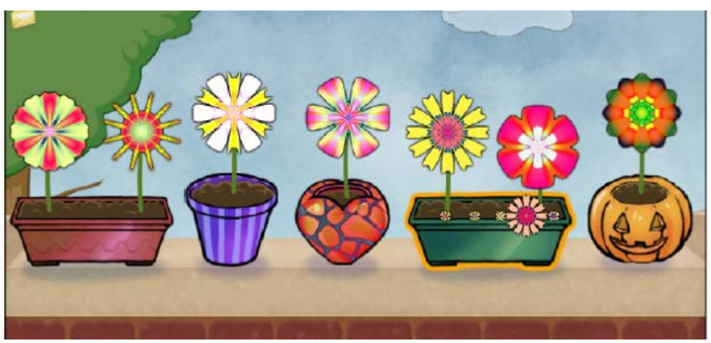
- Use evolutionary computation to search the design space for good artifacts (e.g. levels)
  - Technically, we could use other stochastic search / optimization algorithms
- Major issues:
  - Representing the content
  - Devising a good evaluation / fitness function

Julian Togelius, Georgios N. Yannakakis, Kenneth O. Stanley and Cameron Browne (2011): Search-based Procedural Content Generation: A Taxonomy and Survey. IEEE TCIAIG.

Search-based Procedural Content Generation

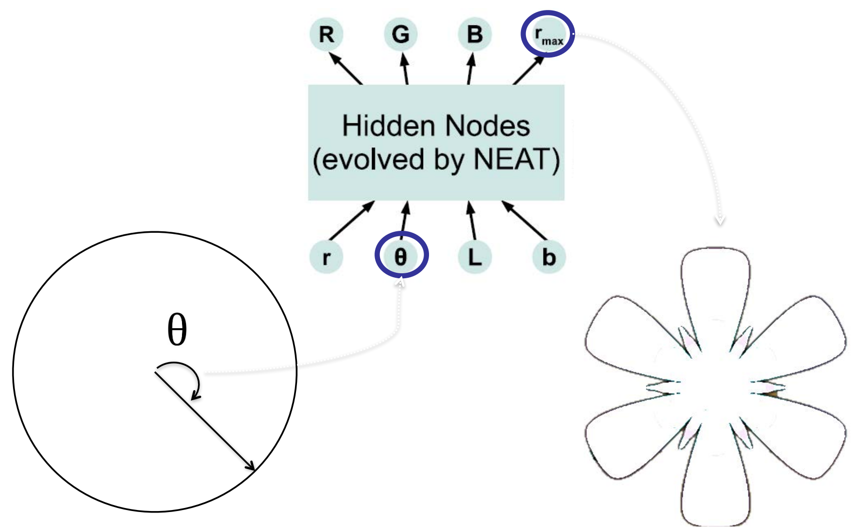


Petalz Social Facebook Game based on PCG through NE

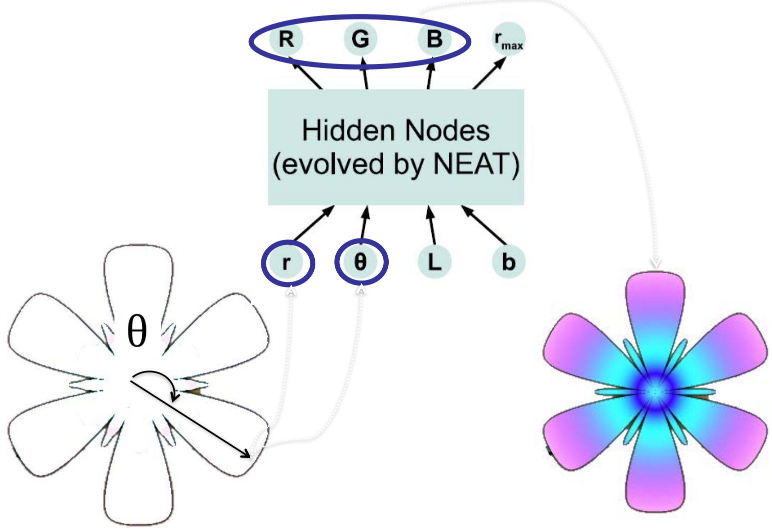


Sebastian Risi, Joel Lehman, David D'Ambrosio, Ryan Hall, Kenneth Stanley, AIIDE 2012, TCIAIG 2015

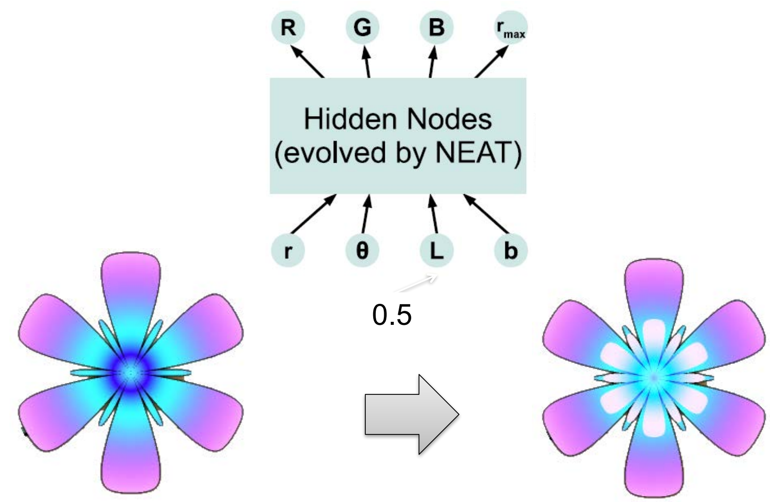
Generating Flower Images and Shapes



Generating Flower Images and Shapes



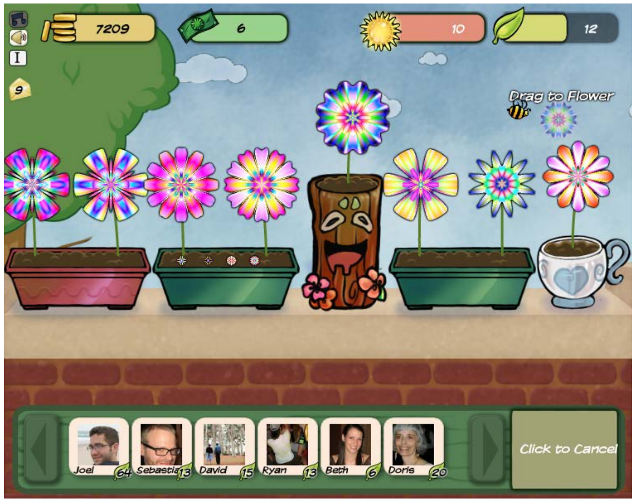
Generating Flower Images and Shapes



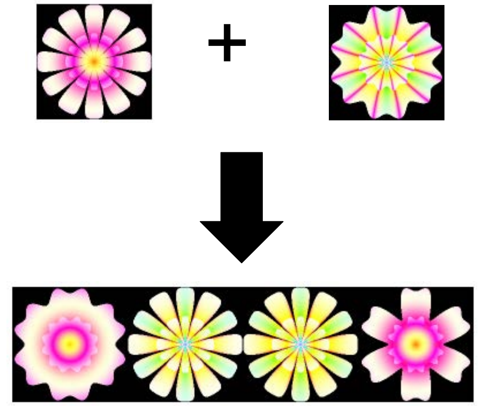
Flower Evolution: Pollinating a Flower

Planting the Offspring

Crosspollination Also Possible



Crosspollination



Hybrid Methods - Latent Variable Evolution (LVE)

- A learned compact genotype-to-phenotype mapping ➔ <u>robust mutations</u>
- Applicable to variety of other

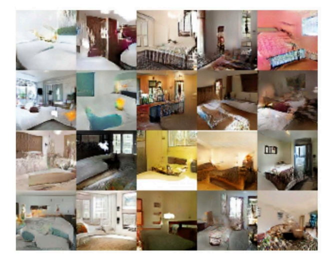
Bontrager, Togelius, Memon 2017

Bontrager, Lin, Togelius, Risi, 2018

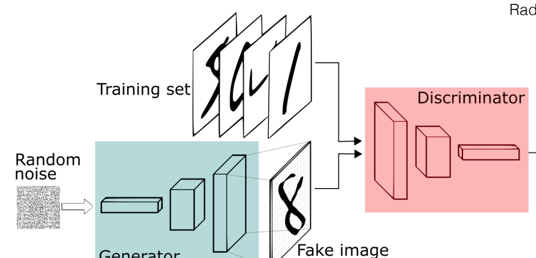
Generative and Adversarial Networks (GANs)

Goodfellow 2014

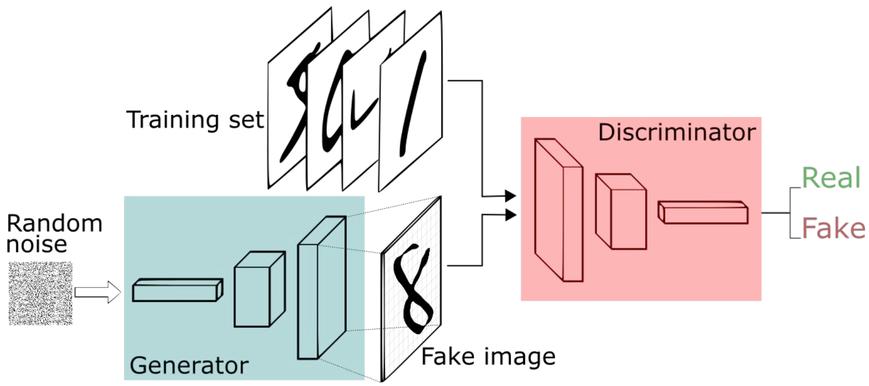
NVIDIA 2017



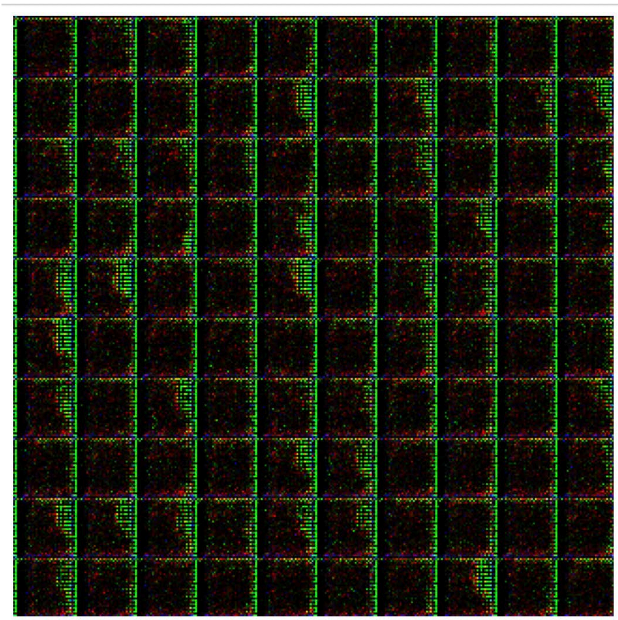
Radford et al. 2015



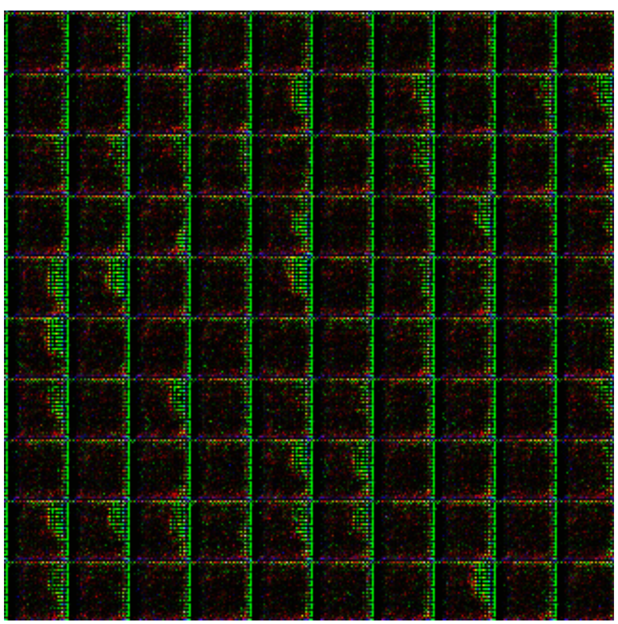
https://deeplearning4j.org/generative-adversarial-network



https://deeplearning4j.org/generative-adversarial-network



https://blog.openai.com/generative-models/



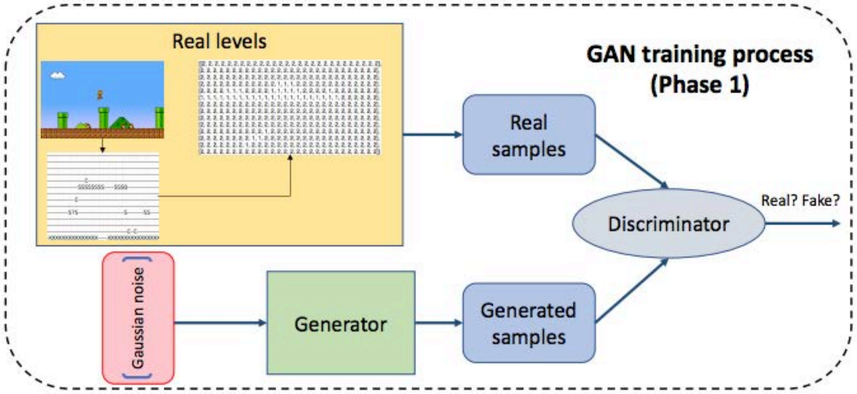
https://blog.openai.com/generative-models/

Real images

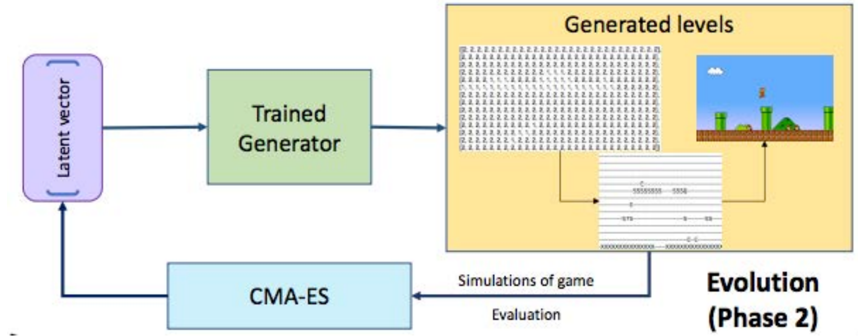
Generated images

https://blog.openai.com/generative-models/

Evolving Mario Levels in the Latent Space of a Deep Convolutional Generative Adversarial Network
Volz, Schrum, Liu, Lucas, Smith, Risi, GECCO 2018



Approach – Phase II



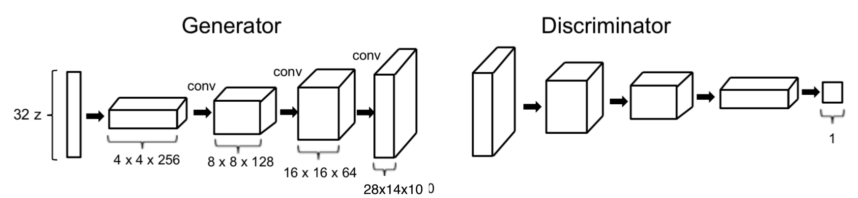
GAN Training

173 training images of size 28x14

Level Representation

Tile type	Symbol	Identity	Visualization
Solid/Ground	X	0	
Breakable	S	1	
Empty (passable)	-	2	
Full question block	?	3	
Empty question block	Q	4	
Enemy	E	5	
Top-left pipe	<	6	
Top-right pipe	>	7	
Left pipe	[	8	
Right pipe	]	9	

GAN changes:
- One-hot encoding
- ReLU activation function for output layer
- Argmax to determine tile type



CMA-ES Experiments

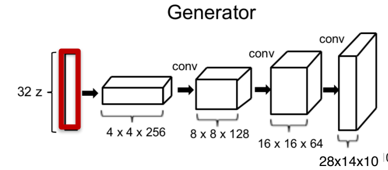
- Representation-based testing:
  - Optimize for certain number of ground titles

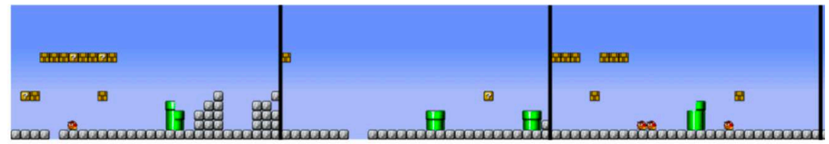
$$F_{ground} = \sqrt{(g-t)^2}$$

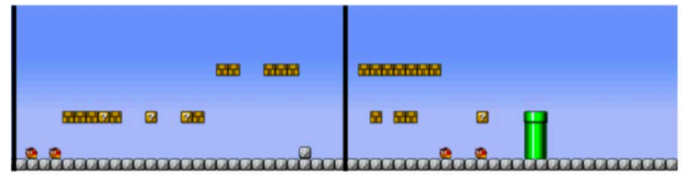
  - Increasing difficulty (less ground, more enemies)

- Agent-based testing:
  A* Mario agent by Baumgarten
  Fitness = %playable + #jumps

Random Sampling

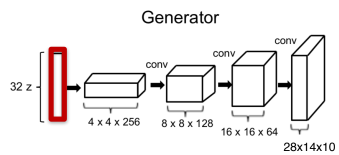


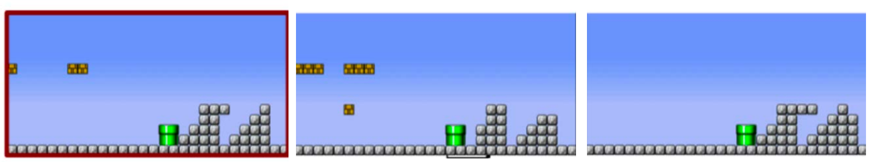




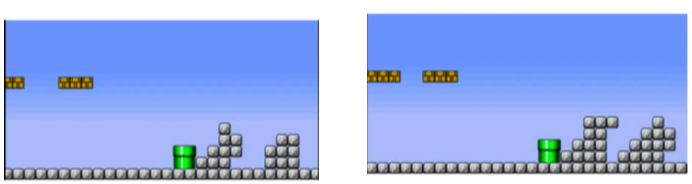
- Trained GAN can express different level variations (can be different to levels used for training)
- Captures domain regularities

Mutations



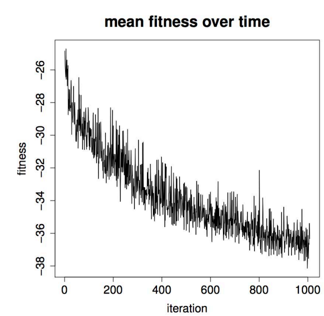


Parent

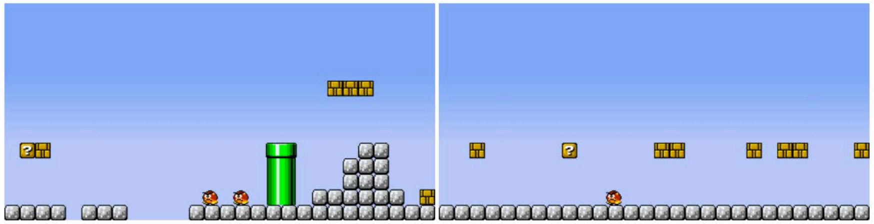


➔ Trained GAN representation displays locality

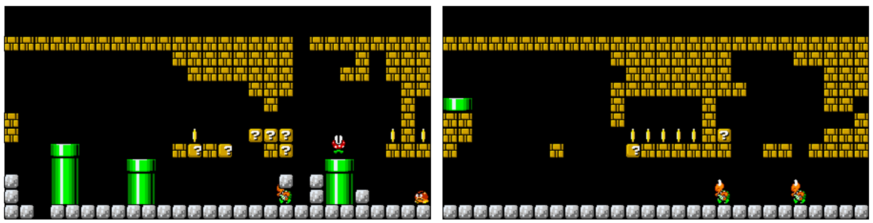
Training



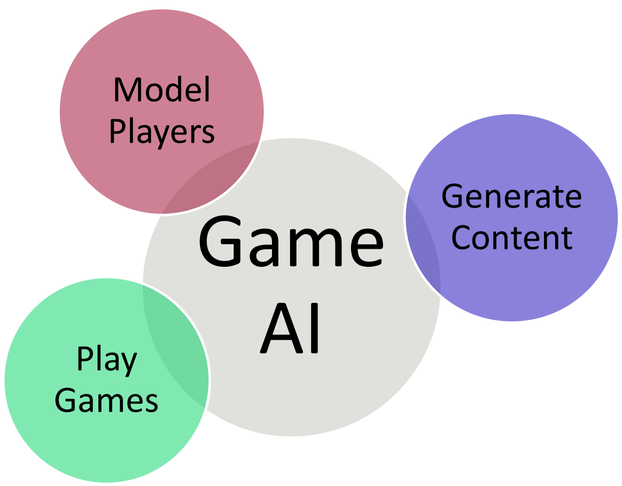
Results



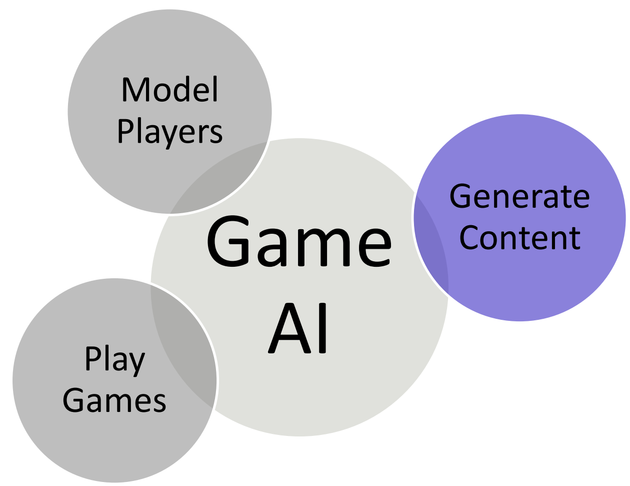
(a) Playable level maximizing jumps (b) Playable level minimizing jumps



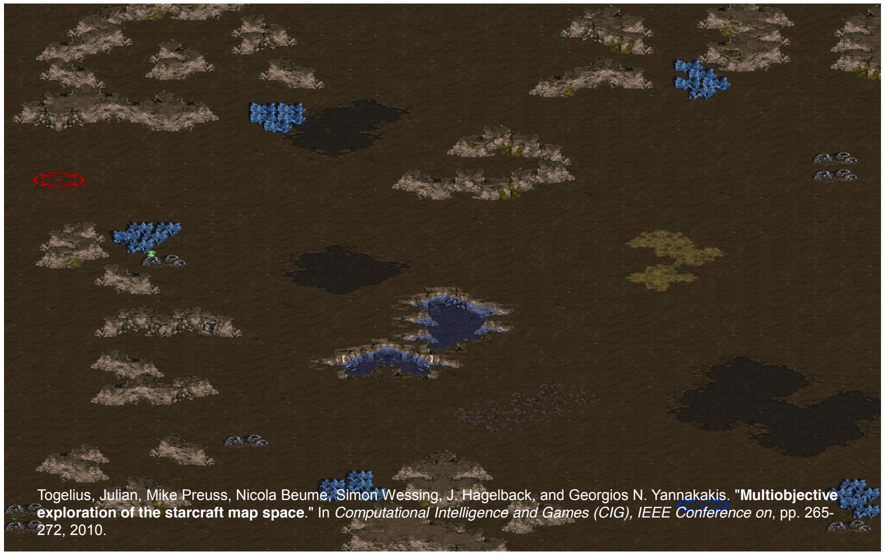
Part II

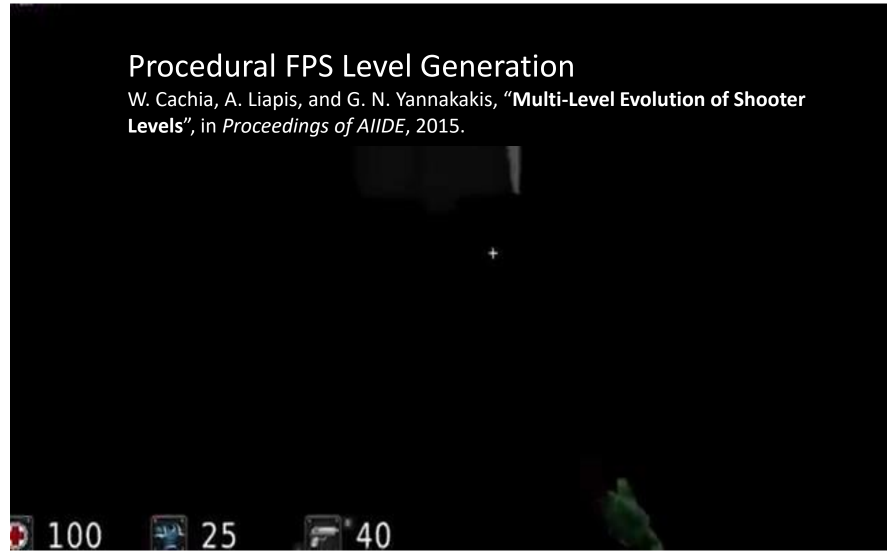


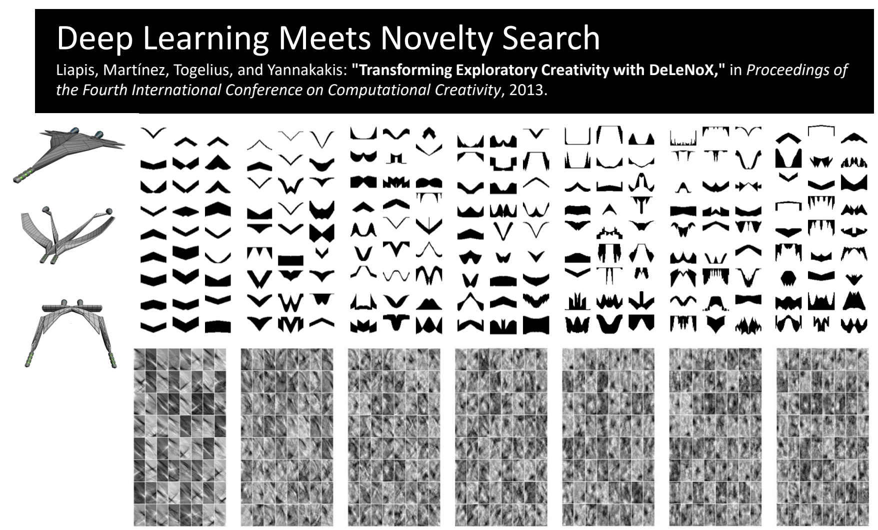
G. N. Yannakakis and J. Togelius, "**Artificial Intelligence and Games**," Springer, 2018.



G. N. Yannakakis and J. Togelius, "**Artificial Intelligence and Games**," Springer, 2018.

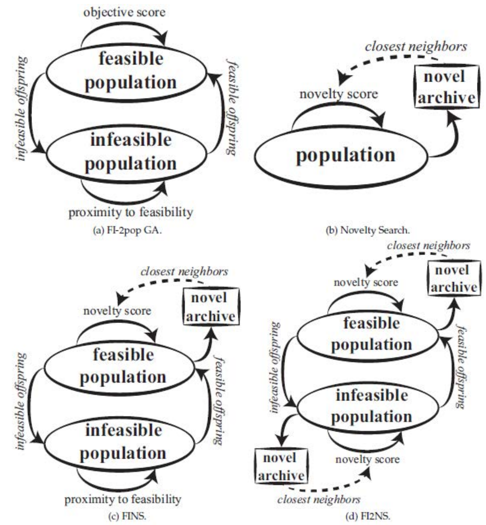

Togelius, Julian, Mike Preuss, Nicola Beume, Simon Wessing, J. Hagelback, and Georgios N. Yannakakis. "**Multiobjective exploration of the starcraft map space**." In *Computational Intelligence and Games (CIG), IEEE Conference on*, pp. 265-272, 2010.


Procedural FPS Level Generation
W. Cachia, A. Liapis, and G. N. Yannakakis, "**Multi-Level Evolution of Shooter Levels**", in *Proceedings of AIIDE*, 2015.
100
25
40

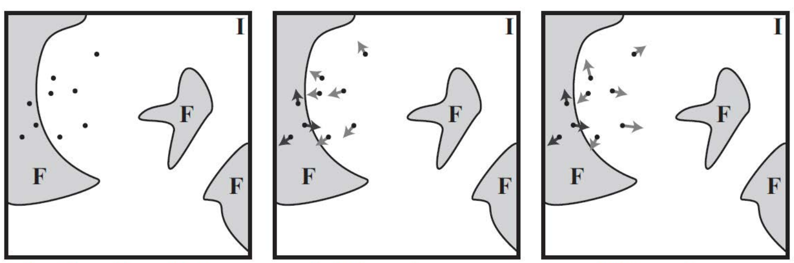

Deep Learning Meets Novelty Search
Liapis, Martínez, Togelius, and Yannakakis: "**Transforming Exploratory Creativity with DeLeNoX,**" in *Proceedings of the Fourth International Conference on Computational Creativity*, 2013.

Antonios Liapis, Georgios N. Yannakakis, Julian Togelius: **"Sentient World: Human-Based Procedural Cartography,"** *EvoMusArt*, 2013.

Constrained Novelty Search

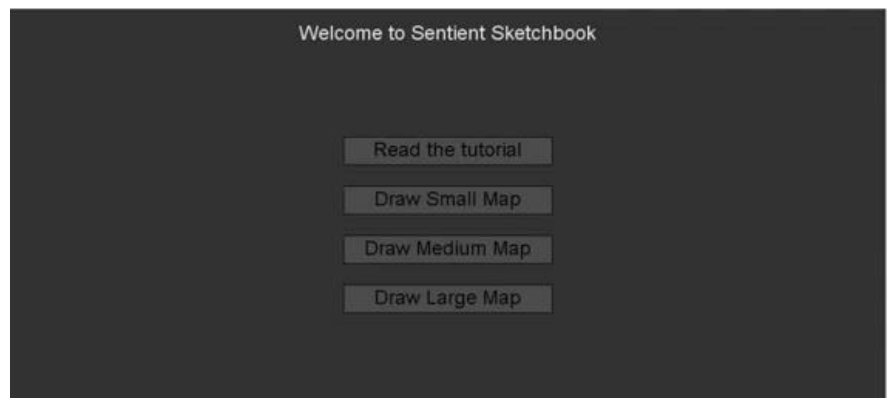


Constrained Novelty Search



(a) Feasible and Infeasible Spaces. (b) FINS search process. (c) FI2NS search process.

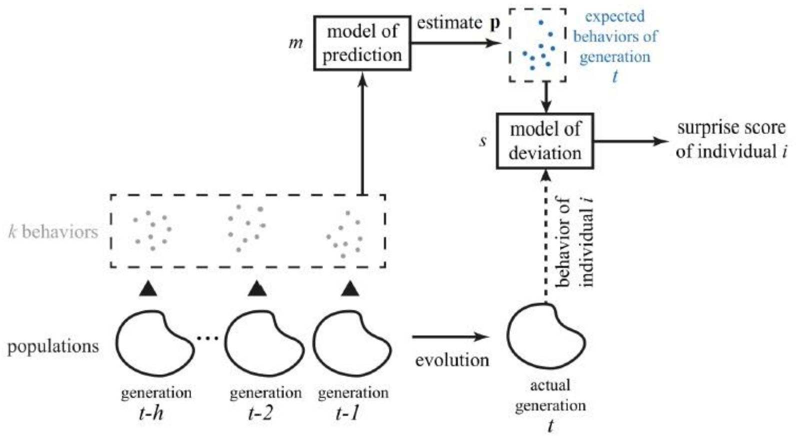
Liapis, Yannakakis and Togelius, **Constrained Novelty Search: A Study on Game Content Generation**, Evolutionary Computation, 21(1), 2015, pp. 101-129



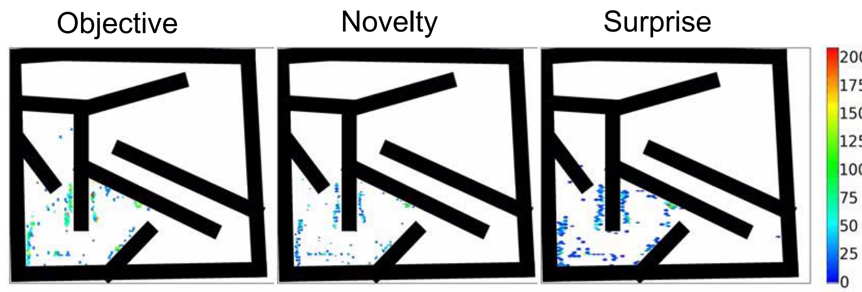
A user can select among a predefined set of map sizes.
Map size determines the number of allowed bases and resources.

Georgios N. Yannakakis, Antonios Liapis and Constantine Alexopoulos: **"Mixed-Initiative Co-Creativity,"** in Proc. of the ACM Conference on Foundations of Digital Games, 2014.

From Novelty Search to **Surprise Search**



Code (C++): http://www.autogamedesign.eu/software

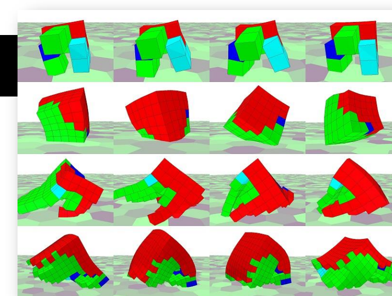


Surprise Search for Problem Solving

Gravina, Liapis, and Yannakakis: **"Surprise Search: beyond Novelty and Objectives"** in Proceedings of GECCO, 2016

Gravina, Liapis and Yannakakis: **"Constrained Surprise Search for Content Generation,"** in Proceedings of the IEEE Conference on Computational Intelligence and Games (CIG). 2016.

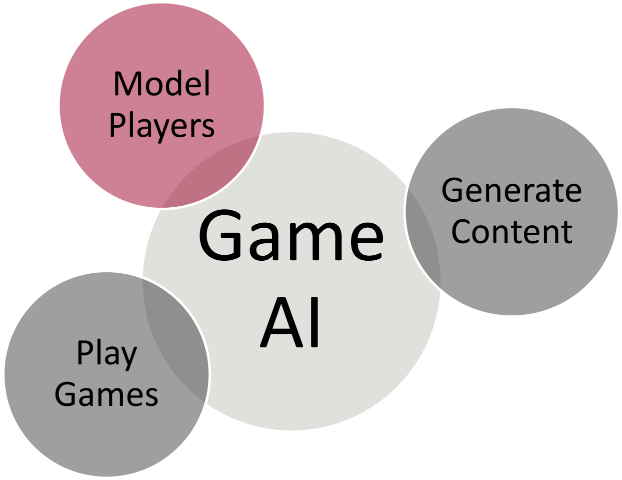
Surprise for QD



- Novelty-Surprise Search: a **robust** and **efficient** divergent search algorithm
  - Maze navigation
  - Robot morphology evolution
- Surprise for **quality diversity**
  - Combined with local competition is highly advantageous

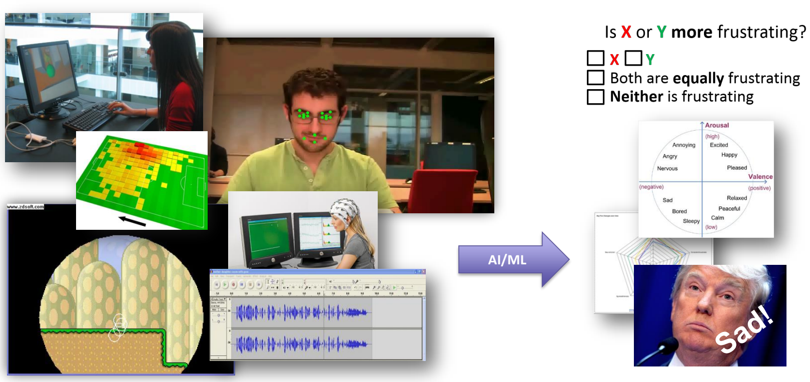
Gravina, Daniele, Antonios Liapis, and Georgios N. Yannakakis. "**Fusing Novelty and Surprise for Evolving Robot Morphologies**" *GECCO* (2018).

Gravina, Daniele, Antonios Liapis, and Georgios N. Yannakakis. "**Quality Diversity Through Surprise**" *arXiv preprint* arXiv:1807.02397 (2018).

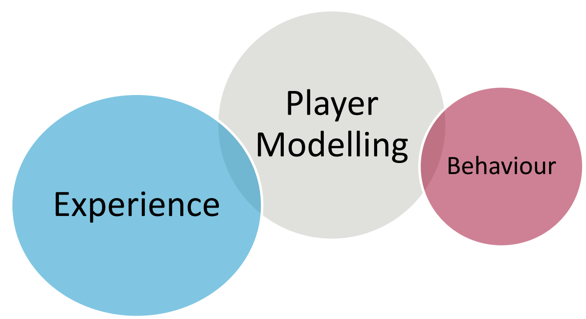


G. N. Yannakakis and J. Togelius, “**Artificial Intelligence and Games**,” Springer, 2018.

How – In a Nutshell



G. N. Yannakakis, P. Spronck, D. Loiacono and E. Andre, “**Player Modeling**,” in Togelius et al., (Eds.) *Dagstuhl Seminar on Artificial and Computational Intelligence in Games*, 2013.

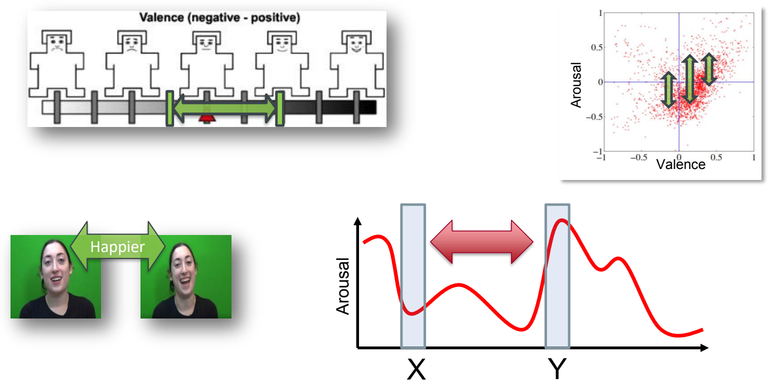


G. N. Yannakakis and J. Togelius, “**Artificial Intelligence and Games**,” Springer, 2018.

Experience: Labels are Key!

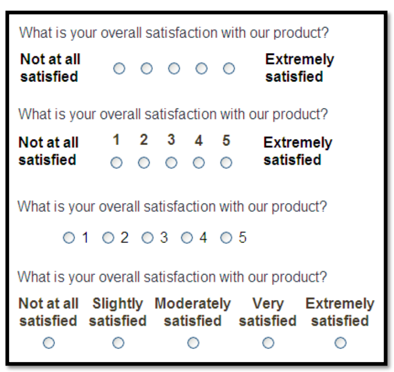
The **ordinal** (relative) approach

Yannakakis, Cowie, Busso, **The Ordinal Nature of Emotions**, ACII, 2017 **[Best Paper Award]**



To sum it up: **Don't** do this!

- Wasteful Info due to
  - Scale-bias
  - Personal-bias
  - Labels are **NOT** numbers
  - High inconsistency (randomness)
  - …


What is your overall satisfaction with our product?

Not at all satisfied ○ ○ ○ ○ ○ Extremely satisfied

What is your overall satisfaction with our product?

Not at all satisfied 1 2 3 4 5 Extremely satisfied

What is your overall satisfaction with our product?

○ 1 ○ 2 ○ 3 ○ 4 ○ 5

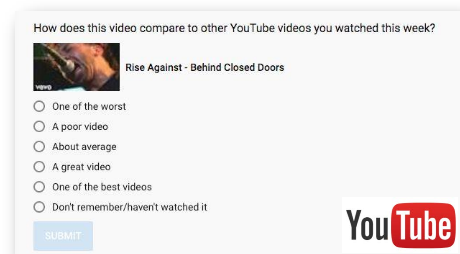
What is your overall satisfaction with our product?

Not at all satisfied	Slightly satisfied	Moderately satisfied	Very satisfied	Extremely satisfied
○	○	○	○	○

Do this instead

- ☐ I like **Julian's** style more/less than **Georgios's** style
- ☐ I like them both equally
- ☐ I like neither

- You gain on
  - Reliability
  - Validity
  - Generality



Modeling **Player Experience**

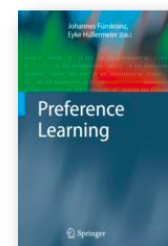
Supervised learning for modelling experience

- **Nominal** values
  - **Julian** is frustrated
- **Numerical** values
  - **Julian** is 0.86 frustrated
- **Ordinal** values
  - **Georgios** is more frustrated than **Julian**

Which Training Method?

Preference learning Classification Regression

Preference Learning



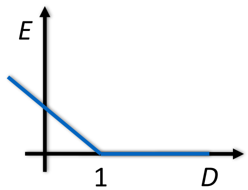
- **Preference learning** is inspired by and built upon humans' limited ability to express their preferences *directly* in terms of a specific (subjective) value function
- Our inability is mainly due to the
  - **subjective nature** of a preference
  - **cognitive load** for assigning specific values to each one of the options
- It is more **natural** to express preferences about a number of options; and this is what we end up doing normally.

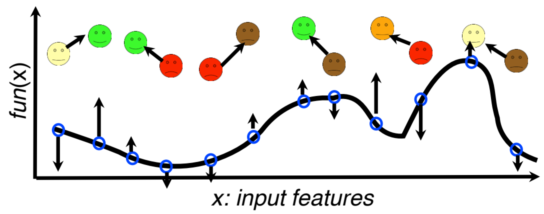
S. Kaci, **Working with preferences: Less is more**. Springer Science & Business Media, 2011.

(Deep) Preference Learning with BP

- Error function maximizes the distance between the output for the preferred sample (d^A) and the output for the non preferred sample (d^B)

$$E = max(0, 1 - (d^A - d^B)) \qquad \frac{\partial E}{\partial w_{ij}} = \begin{cases} -\frac{\partial d^A}{\partial w_{ij}} + \frac{\partial d^B}{\partial w_{ij}} & \text{, if } d^A - d^B < 1 \\ 0 & \text{, otherwise} \end{cases}$$

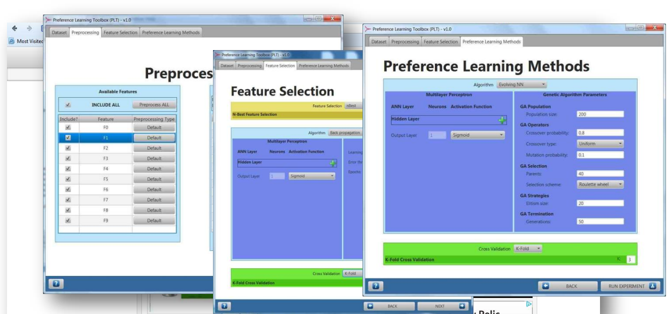




H. P. Martinez, Y. Bengio and G. N. Yannakakis, "**Learning Deep Physiological Models of Affect**," *IEEE Computational Intelligence Magazine*, Special Issue on Computational Intelligence and Affective Computing, pp. 20-33, May, 2013.

An Open-Source Preference Learning Toolbox

Farrugia, Martinez and Yannakakis, **The Preference Learning Toolbox**, arXiv preprint, 2015

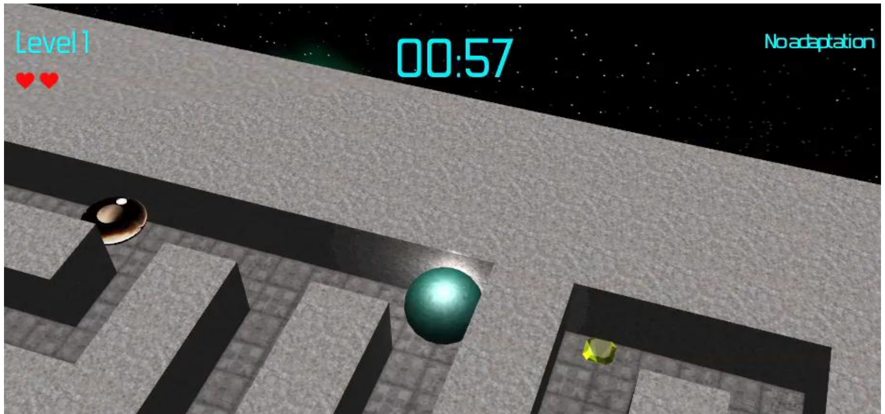


https://sourceforge.net/projects/pl-toolbox/

(Deep) Preference Learning beyond BP

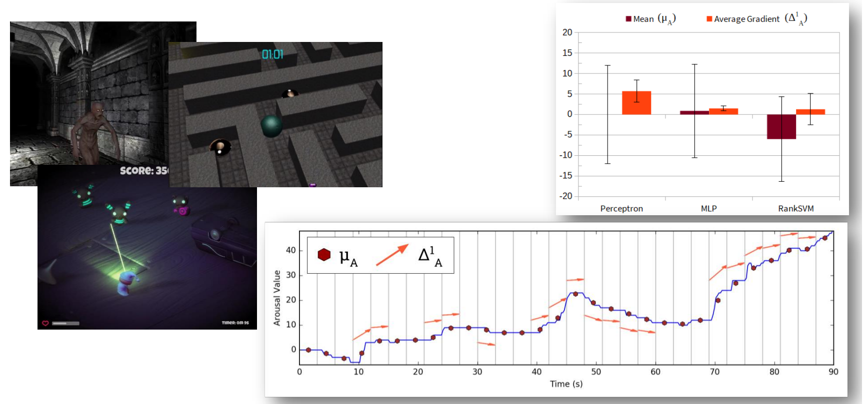
- **Learning from pairs of preferences** can be implemented in most supervised learning methods by adapting the error/fitness function
  - **Neuroevolution**
    - Fitness that rewards match of pairs
  - Rank-based ANN (RankNet)
  - SVMs (RankSVM)
  - Decision Trees
  - …

Some Preference Learning Examples



Emotionally Adaptive Cameras

Yannakakis, Martinez, Jhala, **Towards Affective Camera Control in Games**, *UMUAI*, 2010

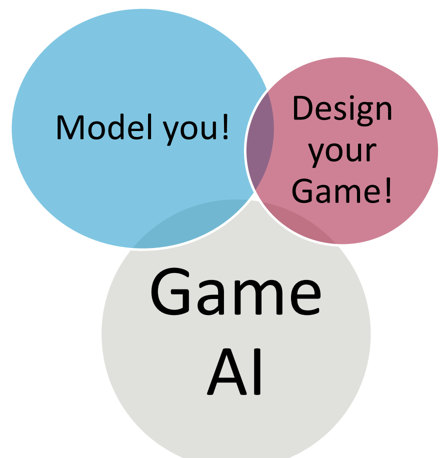


General Models of Affect

Camilleri, Yannakakis and Liapis, **Towards General Models of Player Affect**, *ACII*, 2017

You have a Player Model… so what ?

Experience-driven PCG



Yannakakis, G. N., & Togelius, J. (2011). **Experience-driven procedural content generation**. *IEEE Transactions on Affective Computing, 2*(3), 147-161.

EDPCG: What is it?

"**A framework** for **personalised generation of content** in human computer interaction (in particular in games). It views **(game) content as the *building block* of user (player) experience**"

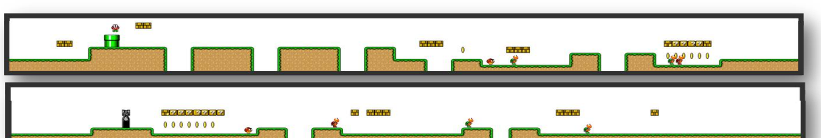
Yannakakis, G. N., & Togelius, J. (2011). **Experience-driven procedural content generation**. *IEEE Transactions on Affective Computing, 2*(3), 147-161.

Experience-driven Level Design in Super Mario Bros

Shaker, Togelius and Yannakakis, **Crowdsourcing the Aesthetics of Platform Games**, *IEEE Trans. on CI and AI in Games*, 2013. **[Outstanding IEEE TCIAIG Paper Award]**

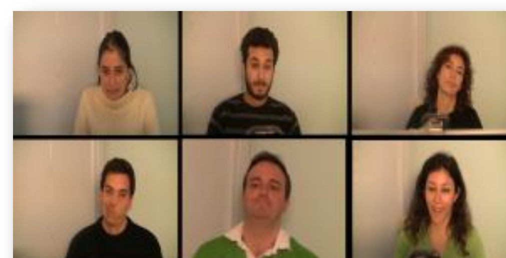
Experience-Driven Level Generation in **Super Mario Bros**

Shaker, Asteriadis, Yannakakis and Karpouzis, **Fusing Visual and Behavioral Cues for Modelling User Experience in Games,** *IEEE Trans. on Systems, Man and Cybernetics* (B), 2013



Platformer Experience Dataset

K. Karpouzis, G. Yannakakis, N Shaker, S. Asteriadis. **The Platformer Experience Dataset**, Sixth Affective Computing and Intelligent Interaction (ACII) Conference, 2015.

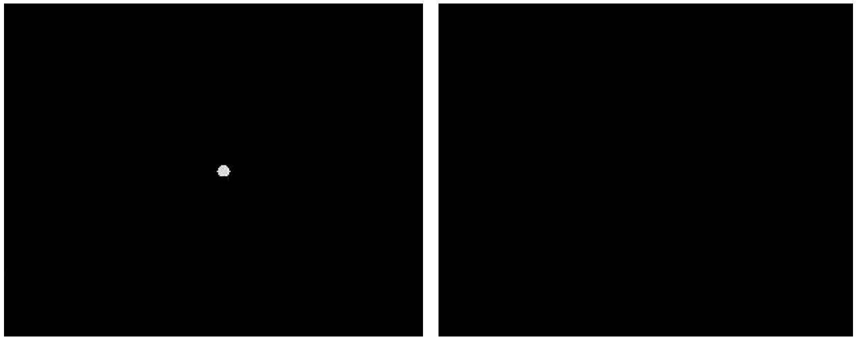


http://ped.institutedigitalgames.com/

Reframing Mario

Game Design for Agent Believability

Camilleri, Yannakakis and Dingli, **Platformer Level Design for Player Believability,** *IEEE CIG*, 2016



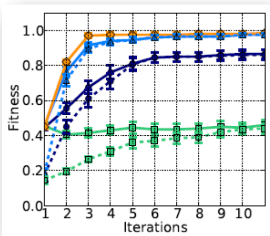
Game Design for Agent Believability

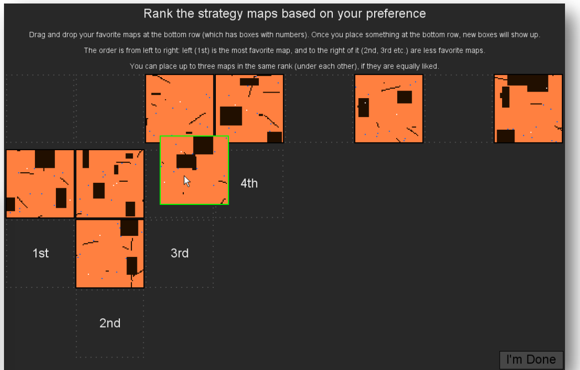
Camilleri, Yannakakis and Dingli, **Platformer Level Design for Player Believability,** *IEEE CIG*, 2016

Player Modeling Beyond **Supervised Learning**

Designer Modeling: Procedural Strategy Map Design

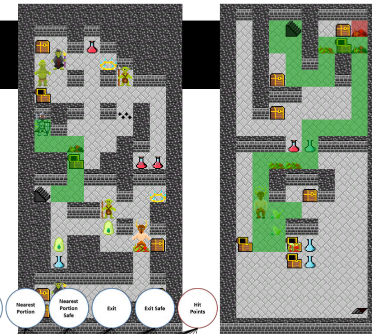
Liapis et al. **Adaptive game level creation through rank-based interactive evolution**. IEEE Conference on Computational Intelligence in Games, 2013.

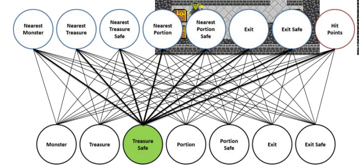




Procedural Personas

- Given utilities (rewards) show me believable gameplay
- Useful for human-standard game testing
- RL
  - MCTS
  - Neuroevolution
  - …
- Inverse RL

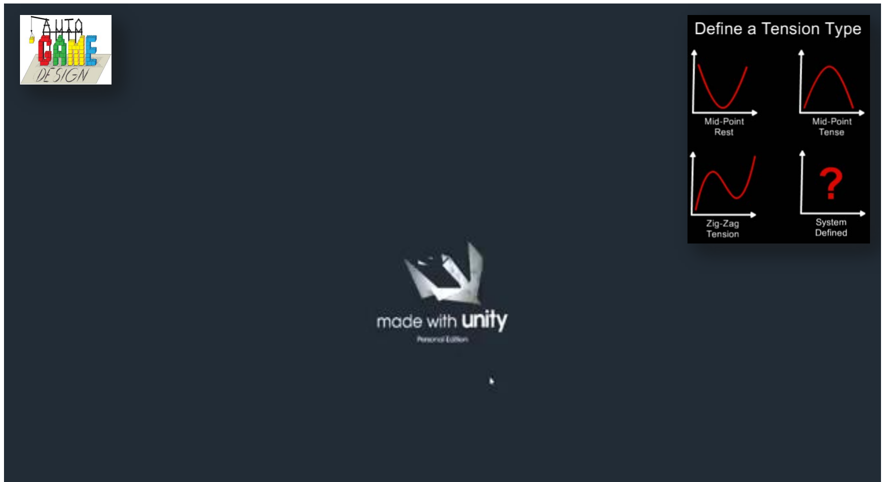




Liapis, Antonios, Christoffer Holmgård, Georgios N. Yannakakis, and Julian Togelius. "**Procedural personas as critics for dungeon generation**." In *European Conference on the Applications of Evolutionary Computation*, pp. 331-343. Springer, Cham, 2015.

Orchestration

Liapis, Yannakakis, Togelius: "**Computational Game Creativity,**" in *Proceedings of the Fifth International Conference on Computational Creativity*, 2014.



Lopes, Liapis, and Yannakakis: **"Sonancia: Sonification of Procedurally Generated Game Levels,"** in Proceedings of the ICCC workshop on Computational Creativity & Games, 2015

“Games: the **final frontier** for AI?”

“AI: the **next step** for Games!”

Julian Togelius, Georgios N. Yannakakis “***General* General Game AI**“ in Proceedings of IEEE CIG, 2016

Get Involved!

IEEE TRANSACTIONS ON

GAMES

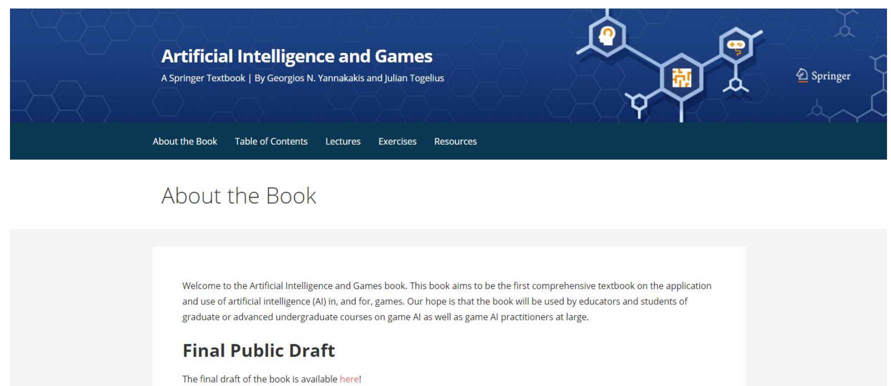
A PUBLICATION OF THE IEEE COMPUTATIONAL INTELLIGENCE SOCIETY,
THE IEEE SENSORS COUNCIL, AND THE IEEE CONSUMER ELECTRONICS SOCIETY

http://cis.ieee.org/ieee-transactions-on-games.html

IEEE Computational Intelligence Society | IEEE Sensors Council

IEEE Computational Intelligence Society

Conference on Computational Intelligence and Games
Department of Data Science & Knowledge Engineering
Maastricht, The Netherlands, August 14-17, 2018



Thank you!

gameaibook.org