CS-E4002 – Special Course in Computer Science:

Seminar on Computational Creativity

Lecture 3: Generation in Computational Creativity

Anna Kantosalo, Christian Guckelsberger & Tapio Takala

Aalto University
School of Science

Agenda

Course practicalities:

- Schedule
- Peer-Reviewing Schedule
- Assignment guidelines
- Finding CC resources

Wrap-up Lecture 2:

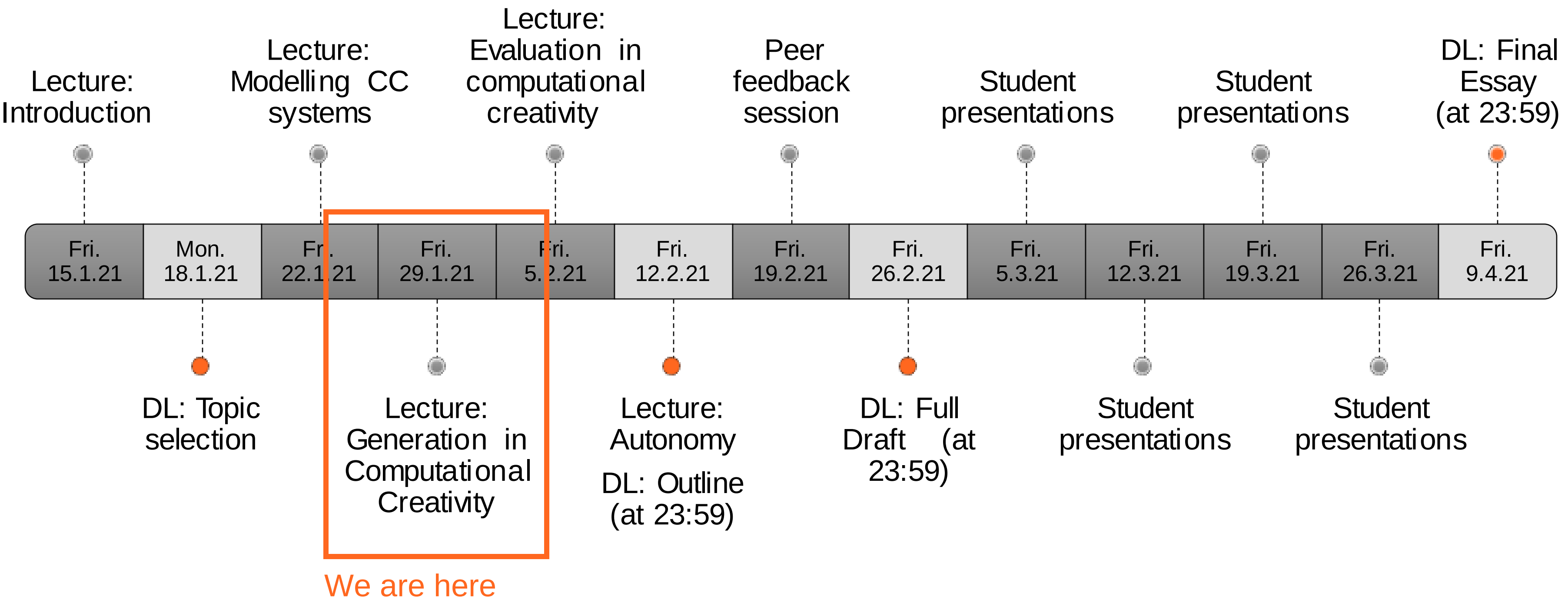
- Creative System Framework Reminder + Exercise
- How to use the CC models

Generation in CC:

- What constitutes “generation”?
- Generation in CC Models
- Generative Algorithms
- Deep-Dive: Genetic Algorithms
- Assignment
- How to use this knowledge of generation for your assessment

Practicalities

Schedule



Aalto University
School of Science

Assignment Guidelines

Essay – Outline DL 12.2.2021

- **2-3 pages**
- **You can use bullet points or short sentences**
- **Focus on the structure of your essay and what is important and interesting about your topic**
- **List your major references**
  - The original reference
  - 2< Domain specific other references
  - Some general references for evaluation and analysis
- **Summarize your topic and analysis well! → You will get better feedback!**

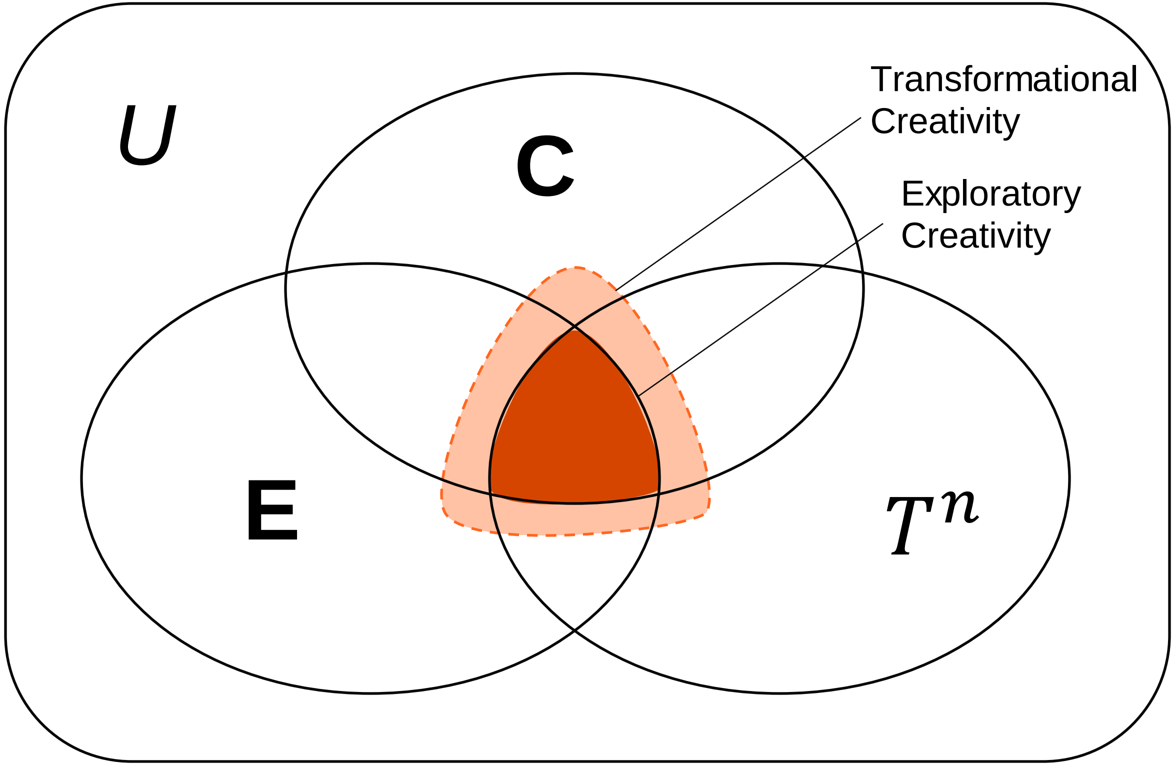
How to find computational creativity resources?

- **Booklists:**
  - http://computationalcreativity.net/home/resources/books/
  - http://computationalcreativity.net/home/resources/bibliography/
- **A list of journals:** http://computationalcreativity.net/home/resources/journals/
- Main conferences:
  - International Conference on Computational Creativity http://computationalcreativity.net/home/conferences/
  - ACM Creativity & Cognition https://dl.acm.org/conference/c-n-c/proceedings
  - There are also several other AI conferences that have published CC related studies or held workshops about it, see e.g. AISB, and AAAI conferences

Wrap-up Lecture 2: Computational Creativity Models

The Creative Systems Framework (Repetition)

- Wiggins' (2006a,b) formalization of Boden's (1990) model: creativity as search in a "conceptual space"
- *U* – The universe of all possible concepts
- *C* – Valid concepts
- *E* – High quality concepts
- T^n – Concepts reachable in n generative iterations



Let’s practice!

For the given domain sample, describe a creative agent by describing

a) The universe

b) The rules for valid concepts

c) The rules for evaluating concepts

d) The rules for traversing the conceptual space

https://miro.com/app/board/o9J_lY55L1g=/

How to use knowledge on CC models for your assignment?

- Consider what are the most useful models for **describing** (the systems in) your essay topic
- **Analyse** the example systems using the models
- Not all models might fit your work, or your work may not fit all parts of a model – it is also interesting **which parts do not fit**, and why.

Lecture 3: Generation in Computational Creativity

Aalto University
School of Science

What constitutes “generation” in Computational Creativity?

What constitutes “generation” in CC?

- **Generation in CC**: producing / driving a creative artefact / process
- But **which components** does a CC system need to realise such creativity?

our_cc_program.py
```python
def generate():
    # What do we need?

```

- Different notions of creativity render ”generation” in CC **ambiguous**:
  - **Psychological, modern**: **novelty** + **value**
    (e.g. ”standard definition of creativity”, lec. 1 / Runco & Jaeger, 2012)
  - **Natural, Pagan**: **novelty-only / value-free.** The unfolding and dissolution of natural processes (cf. Still & d’Inverno, 2016).
- Does ”generation” in CC only produce novelty, or also evaluate?

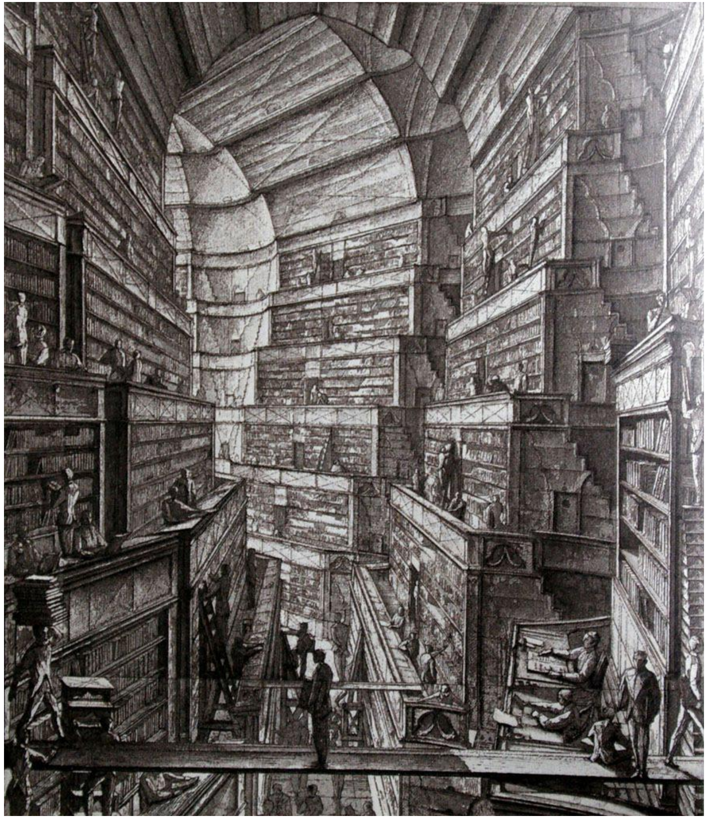
What constitutes “generation” in CC?

Generative creativity: “an instance of a system creating new patterns or behaviours regardless of the benefit to that system. There is an explanation for the creative outcome, but not a reason.”

-Bown, 2012

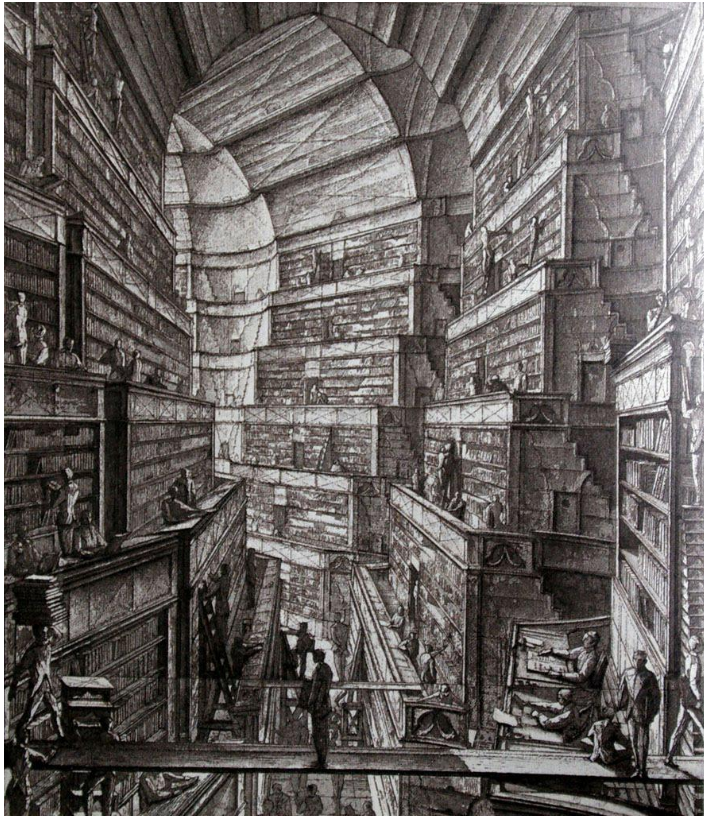
The need for evaluation

- But: for CC goals (lecture 1), we need value assessment!
- Example: generate **books** with ..
  - 410 pages, 40 lines per page, 80 characters each
  - **Variety** through 25 basic characters (22 letters, the period, the comma, the space)

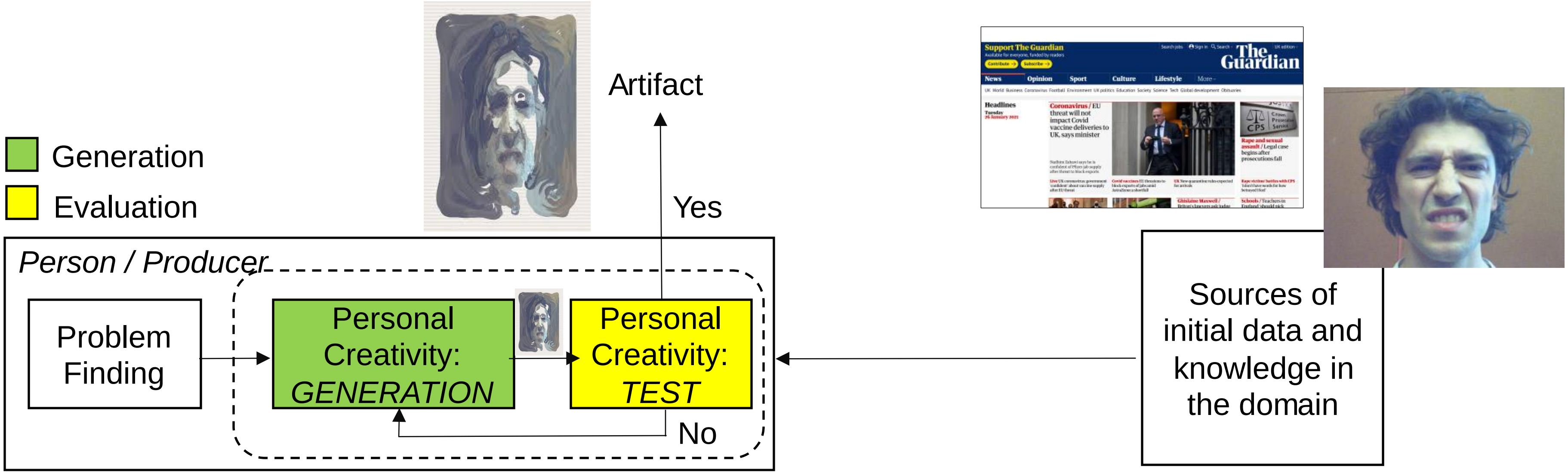


The need for evaluation

- **The Library of Babel** (1941) by Jorge Luis Borges (1899–1986)
- How big is the space of books?
  - 410 x 40 x 80 = 1,312,000 positions per book
  - Each can be filled in 25 ways
  - $25^{(410x40x80)} = 25^{1,312,000}$ books!
- Cf. 10^{78} atoms in universe!
- Evaluation needed to **filter** artefacts & not overwhelm human user!

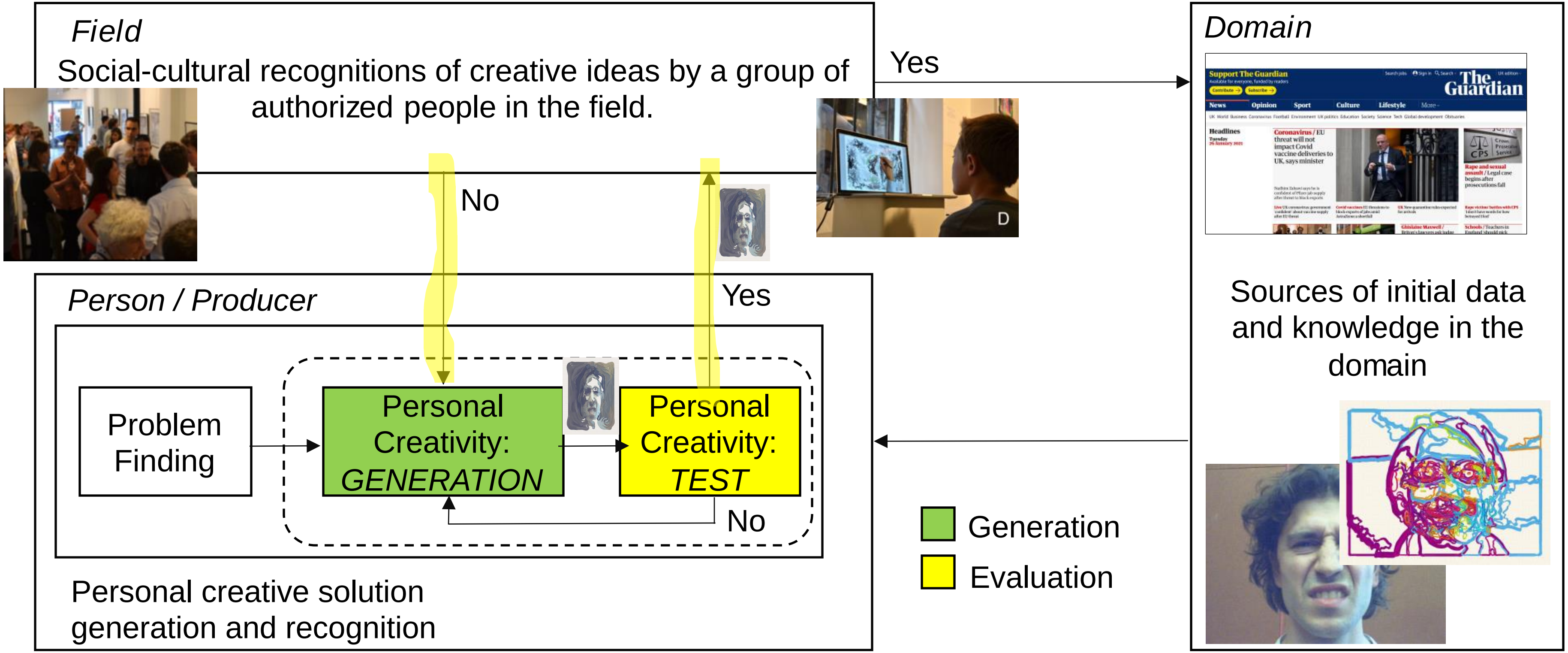


Models of (human) creativity: Generate-and-Test



- Evaluation als needed to **model human creativity**!
- Running example: **portrait generation** (e.g. Colton et al. 2015)

Models of (human) creativity:
Dual Generate-And-Test (Liu, 2000)



Summary: the need for evaluation

We require the assessment of artefact value to:

- **Filter** artefacts & not overwhelm human user (Saunders, 2009)
- Model (**human) creativity** (cf. CC goals, lec. 1; and Colton, 2008)
- Design systems that **appear** more **creative** (lec. 4; Jordanous, 2012)
- Allow for more creative **autonomy** (lec. 5; Jennings, 2010)

Therefore, in most CC systems, **generation entails evaluation, or:**

“generation” = value-less generation + evaluation

Break (5 mins)

Generative Algorithms

Generative Algorithms

- There's many different ones …

Generative Adversarial Networks

Cellular Automata

Evolutionary Algorithms

Reinforcement Learning

Fractals / Graphtals

Constraint solvers

(Probabilistic) Context-Free / Context-Based Grammars

…

(Variational) Autoencoders

(Hidden) Markov Chains

Recurrent Neural Networks

Transformers

Rule-bases systems

Bayesian Networks

- Evaluation can be **explicit** (objective/fitness/cost function, etc.), **implicit** (via rules) or **learned from data** (generative models)

Generative Algorithms

- There's many different ones …

Generative Adversarial Networks

Cellular Automata

Evolutionary Algorithms

Reinforcement Learning

Fractals / Graphtals

(Probabilistic) Context-Free / Context-Based Grammars

Constraint solvers

…

(Variational) Autoencoders

(Hidden) Markov Chains

Recurrent Neural Networks

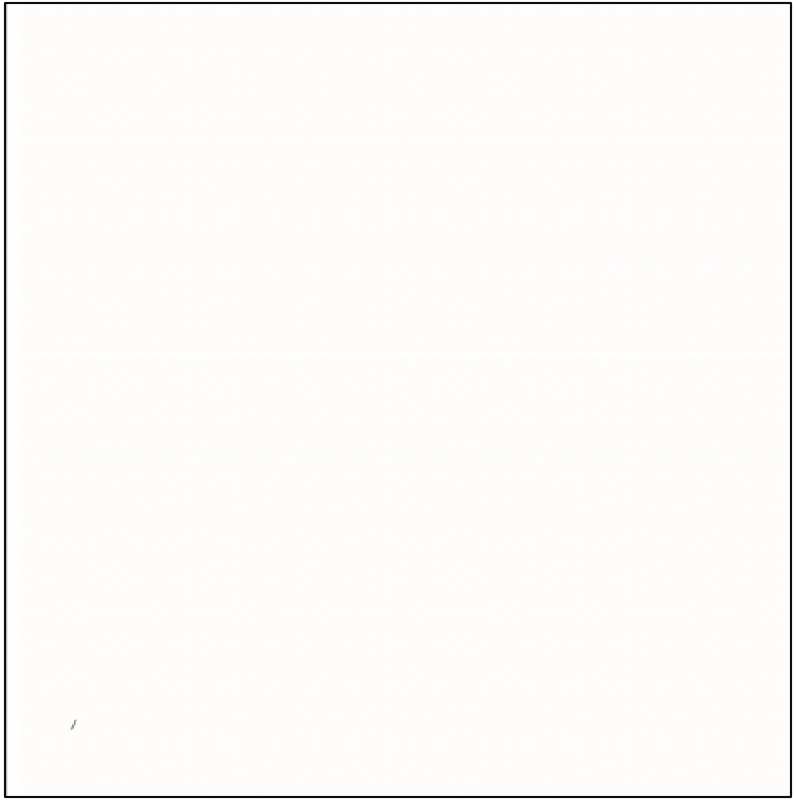
Transformers

Rule-bases systems

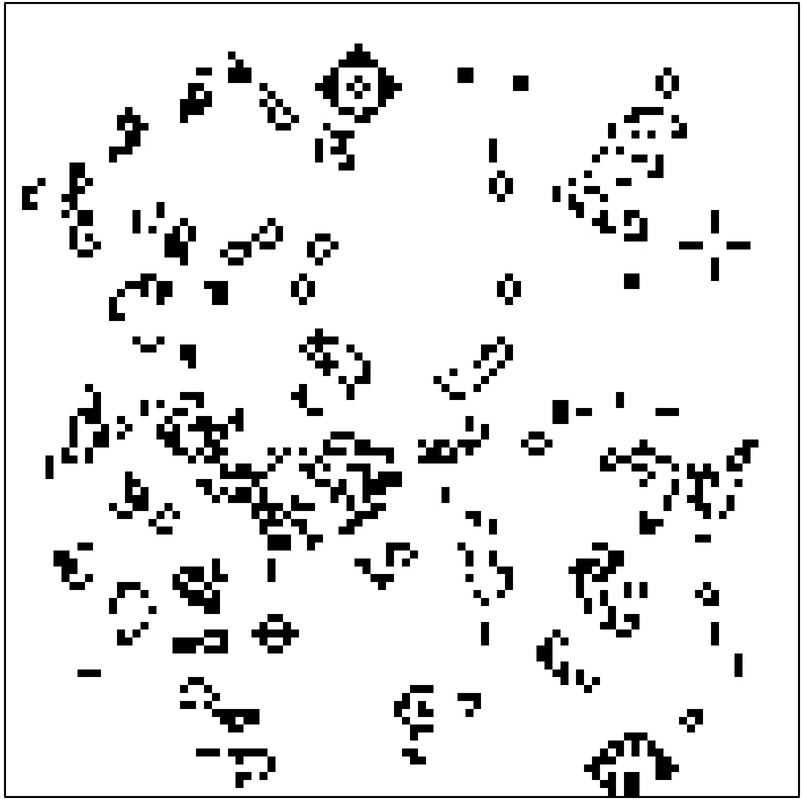
Bayesian Networks

- Evaluation can be **explicit** (objective/fitness/cost function, etc.), **implicit** (in rules) or **learned from data** (generative models)

… in computational value-free creativity

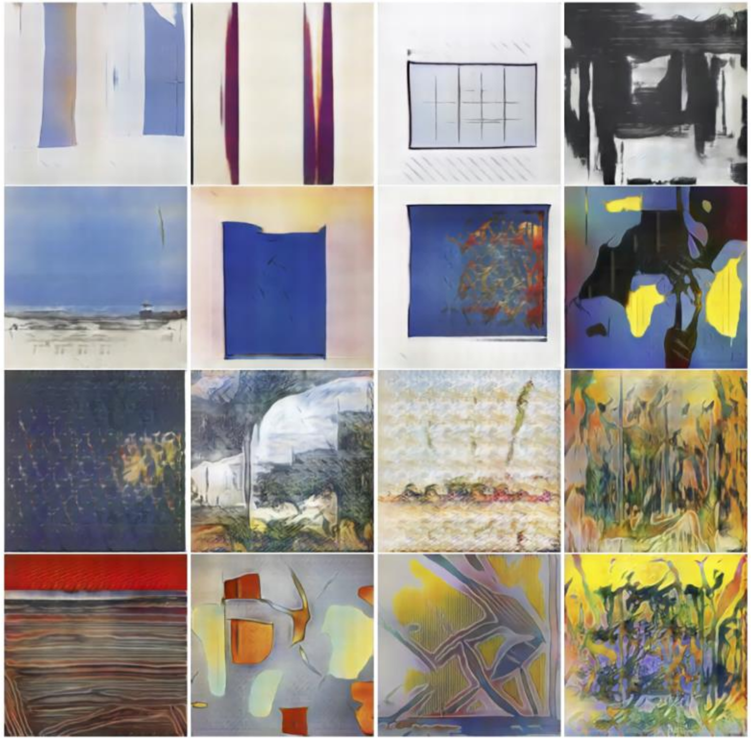


Weed (Lindenmeyer System)



Conway’s Game of Life (Cellular Automata)

… in computational value-full creativity



Creative Adversarial Networks
(Elgammal et al., 2017)

The 2006 NASA ST5 spacecraft antenna
(Hornby et al., 2006)

Deep Dive: Genetic Algorithms

Example Domain: Videogame Procedural Content Generation

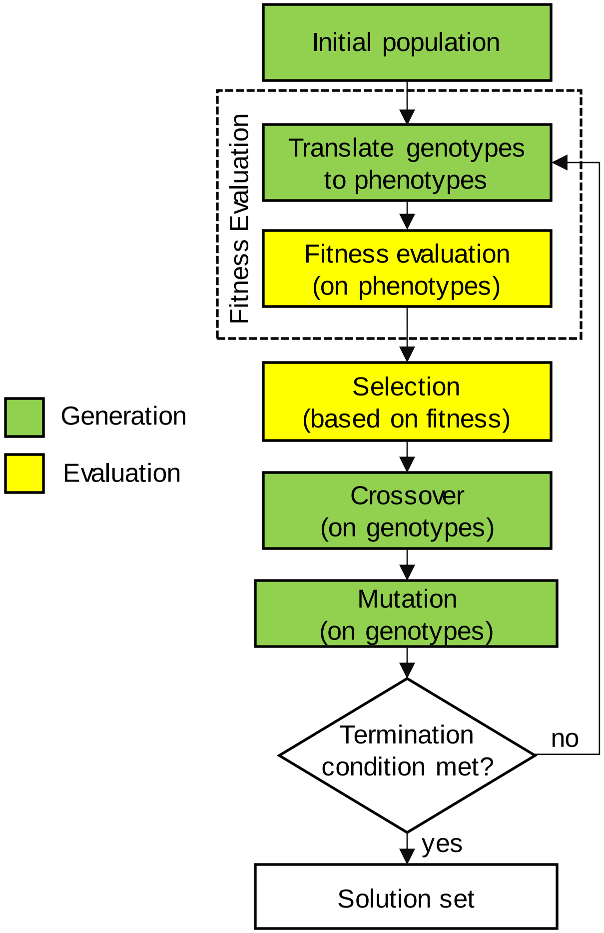
Genetic Algorithms

- Heuristic optimisation algorithm (Holland 1975/92) < evolutionary algorithms
- Random exploration & information exchange of parallel, climbing search threads
- Inspired by evolution – survival of the fittest:
    - Evolving a **population** of **individuals** over multiple **generations**
    - Each individual = **genotype**, expressing a **phenotype**

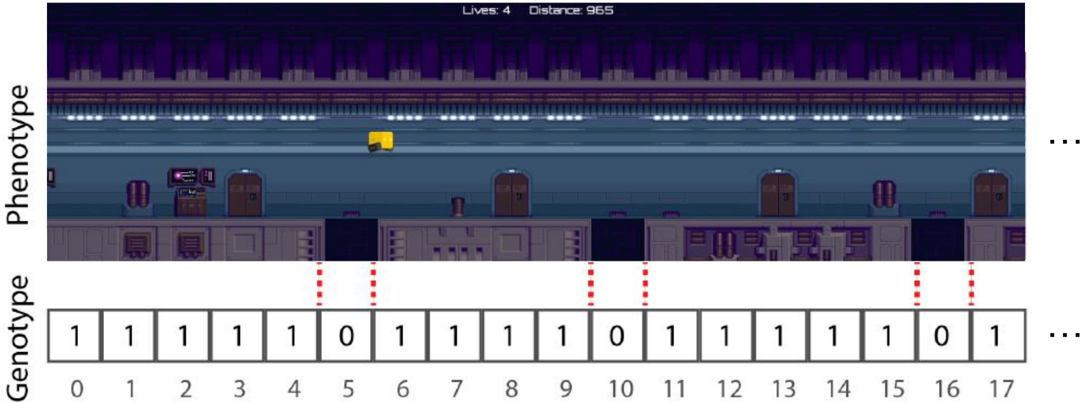
The 2006 NASA ST5 spacecraft antenna (Hornby et al., 2006)

Genetic Algorithms

- Central steps:
  1. **Evaluation:** assessing **fitness** of individuals in a generation via their genotypes. **Translation** can be complex.
  2. **Selection**: **sampling** individuals in **next generation** based on their fitness
  3. **Breeding** new individuals: **mutation** and **crossover** of genotype
  4. **Termination** after n generations or if fixed fitness threshold reached
- Generation < fitness evaluation if phenotype = output of generator parametrised by genotype



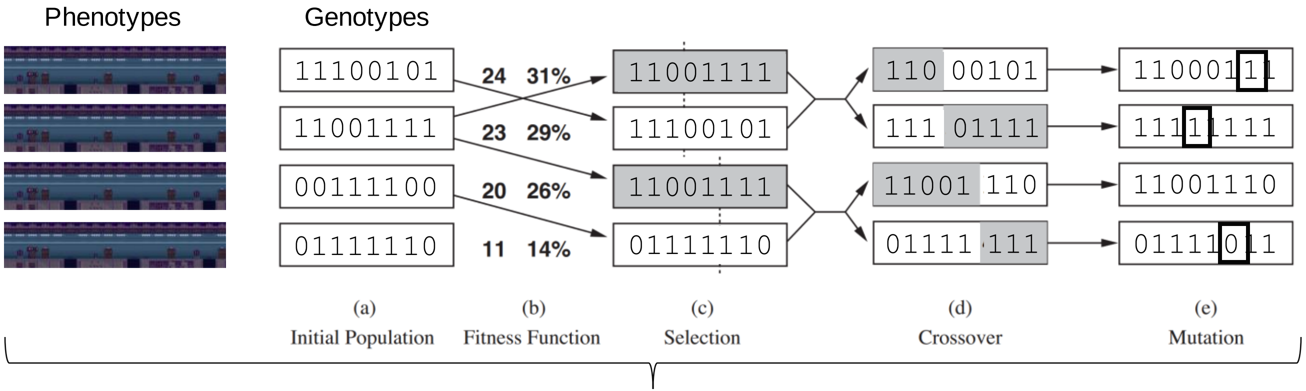
Genetic Algorithms: Example



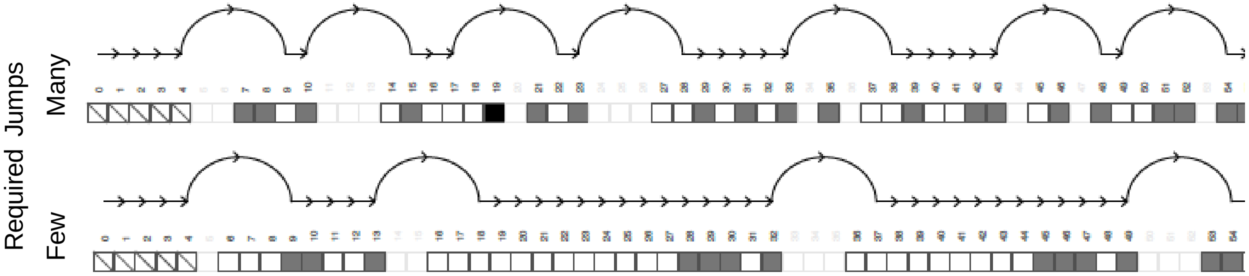
- E.g. evolving game levels for higher difficulty (cf. Guckelsberger et al., 2017)
- **Genotype-Phenotype mapping**: bitarray, 1=platform, 0=gap
- **Fitness assessed using simulations of "lazy" AI player:**
  - no. required jumps to goal;
  - 0 if no path from start to end of level.

  } normalised by max. jumps / generation

Genetic Algorithms: Example



Individuals in **generation 300** (separate optimisation for many / few jumps):



Genetic Algorithms: Choosing a good genotype-phenotype mapping (Togelius et al., 2011)

- **Genotype** representations: vector of bits, natural / real numbers; graphs, …
- Limit **length**: Genotype should be capable of representing the content an all possible solutions; but not too long to become intractable.
- Ensure high **locality**: small change in genotype -> small change in phenotype -> small change in fitness
- Genotype-Phenotype Mappings are on a **continuum** between:
  - **Direct encodings**: genotype size = linearly proportional to phenotype size; each part maps to specific part of phenotype; simple computation.
  - **Indirect encodings**: sizes can be nonlinearly and unproportional; often involves complex computation.

Genetic Algorithms: Choosing a good genotype-phenotype mapping (Togelius et al., 2011)

indirect representation. As a concrete example, a maze (for use e.g. in a "roguelike" dungeon adventure game) might be represented:

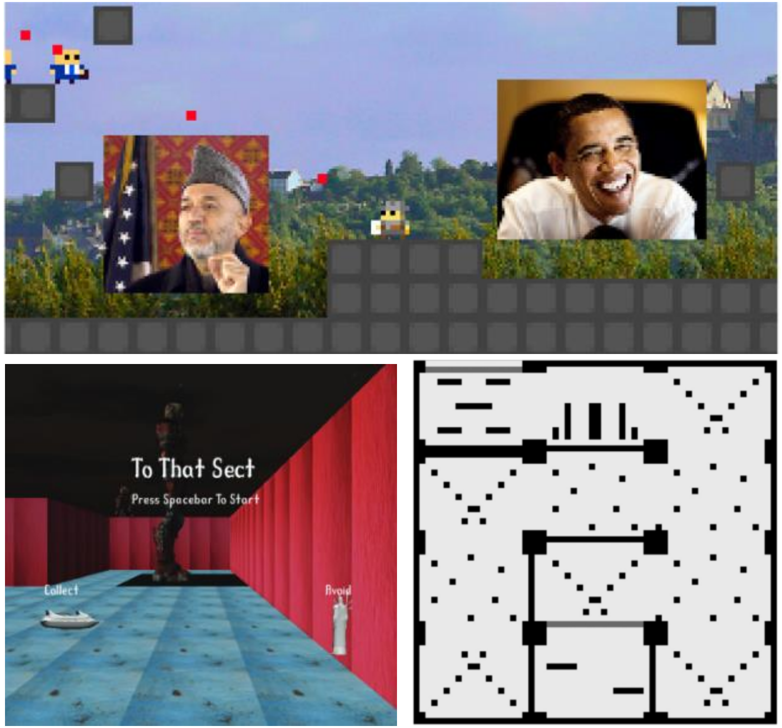
1) directly as a grid where mutation works on the contents (e.g. wall, free space, door, monster) of each cell,
2) more indirectly as a list of the positions, orientations and lengths of walls (an example of this can be found in [29]),
3) even more indirectly as a repository of different reusable patterns of walls and free space, and a list of how they are distributed (with various transforms such as rotation and scaling) across the grid,
4) very indirectly as a list of desirable properties (number of rooms, doors, monsters, length of paths and branching factor), or
5) most indirectly as a random number seed.

These representations yield very different search spaces.

Dwarf Fortress Steam Edition (2020)

Genetic Algorithms: Types of Fitness Functions (Togelius et al., 2011)

1. **Direct:** calculated on features extracted from phenotype. Typically little computation.
2. **Simulation-based:** calculated based on features extracted from the simulated interaction of artificial agent with artefact. Computation can be complex.

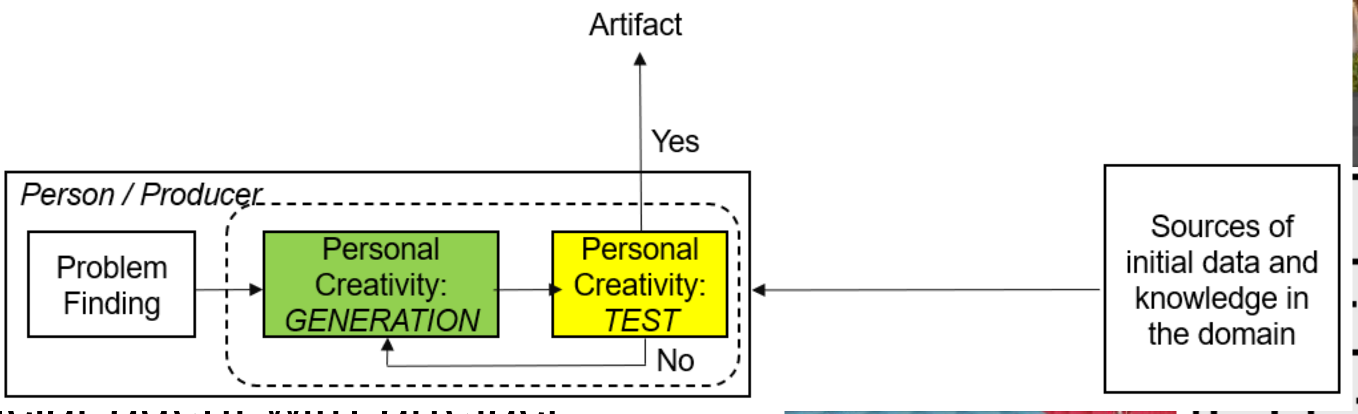


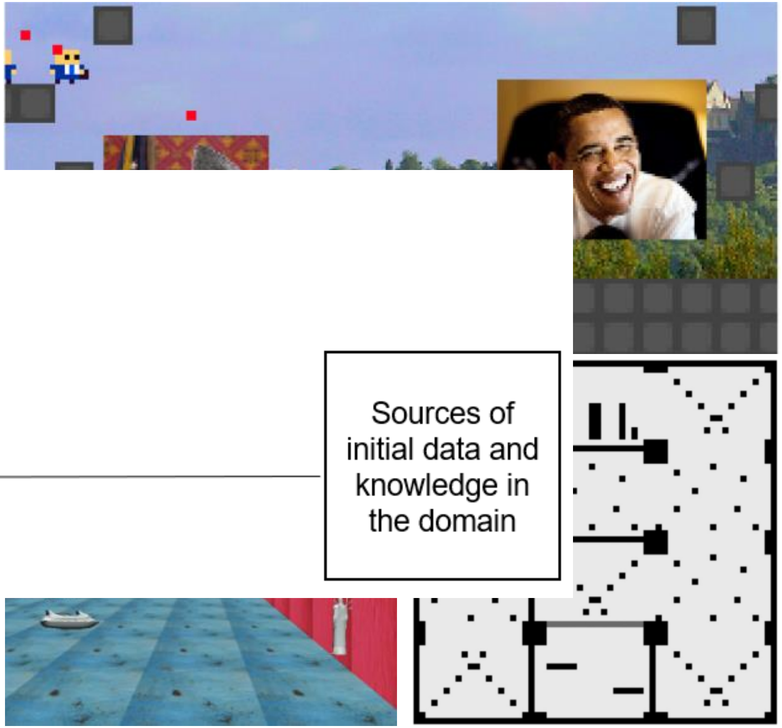
ANGELINA Game Design System
(Cook, Colton and Gow, 2016a,b)

Aalto University
School of Science

Genetic Algorithms: Types of Fitness Functions (Togelius et al., 2011)

1. **Direct:** calculated on features extrac
   Typica
2. **Simul**
   based
   from t
   of artif

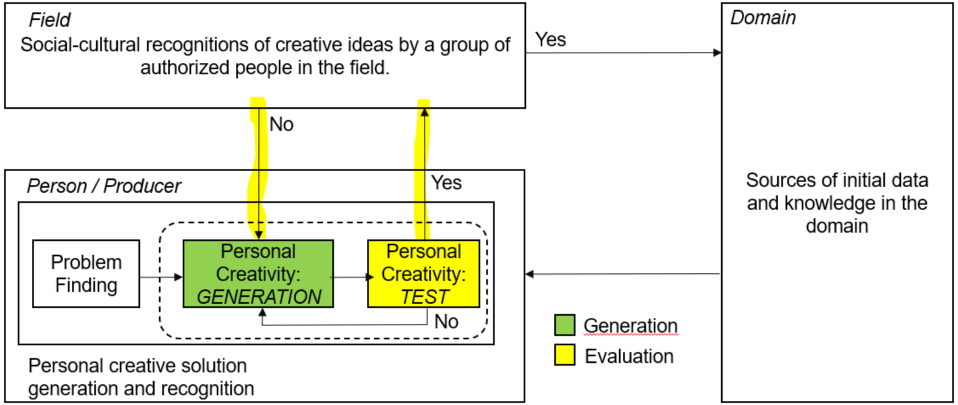


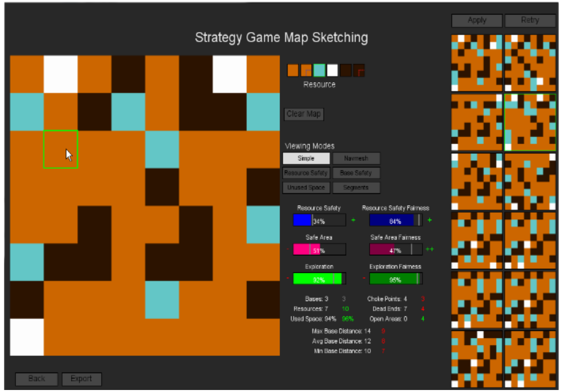


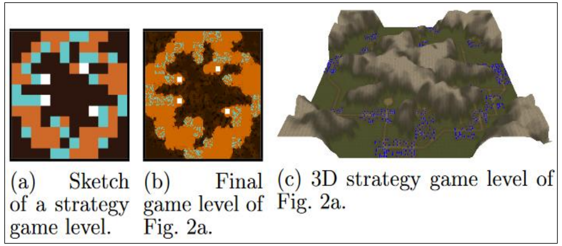
ANGELINA Game Design System
(Cook, Colton and Gow, 2016a,b)

Genetic Algorithms: Types of Fitness Functions (Togelius et al., 2011)

3. **Interactive:** fitness implicitly or explicitly **determined by human** interacting with (partial) artefacts.







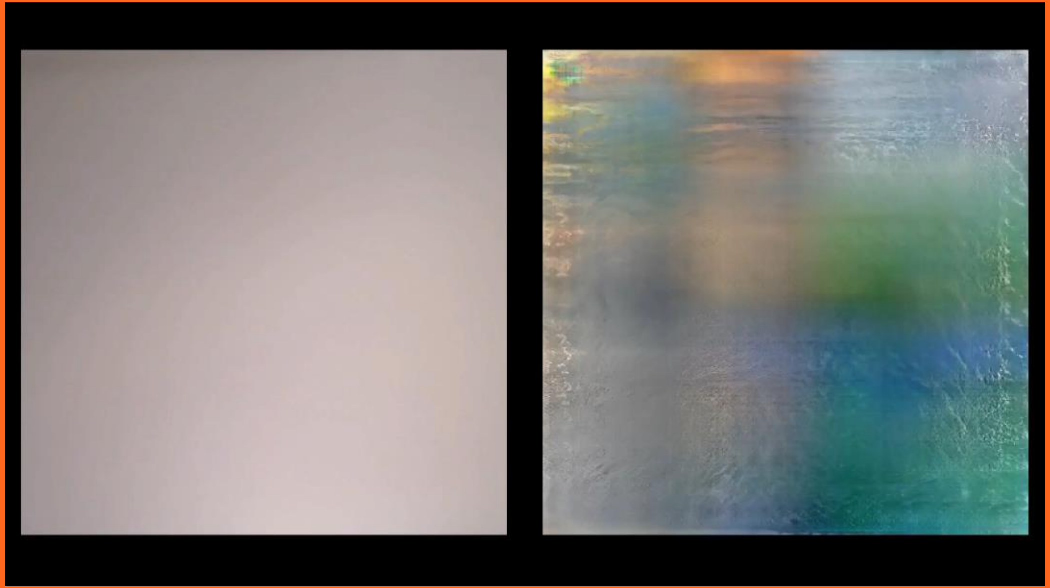
Sentient Sketchbook
(Yannakakis, Liapis & Alexopoulos, 2014)

Let’s practice!

- **Following Russel and Norvig (2003), you can try a genetic algorithm if you can answer 4 questions:**
  1. What is the fitness function?
  2. How is an individual represented? (genotype-phenotype mapping)
  3. How are individuals selected?
  4. How do individuals reproduce?
- **Exercise:** For the given domains, **informally** define **1 and 2**!

https://miro.com/app/board/o9J_lW3YlXg=/

Generative Adversarial Networks (GANs, Goodfellow et al., 2014)



Gloomy Sunday, Memo Akten, 2017

StyleGAN2 (Karras et al., 2020)

Good course with test questions:

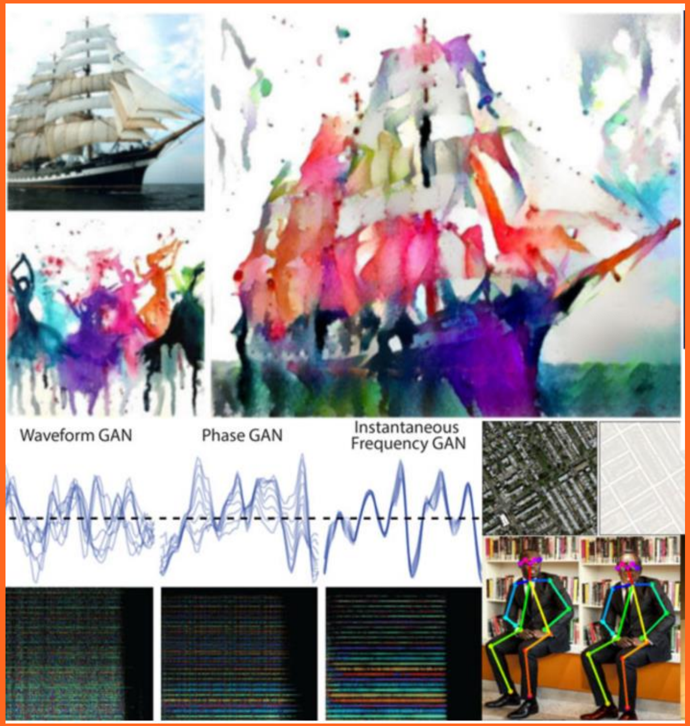
https://developers.google.com/machine-learning/gan

How to use knowledge on generation for your assignment?

- Identify **standard** (but fine-tuned) algorithms in **complex** system architectures
- For a specific CC system, understand how the **creative process** is driven and how artefacts are created
- Given knowledge of different generative algorithms, consider which **weaknesses could be overcome** by replacing the generative and/or evaluation component in a system with an alternative.

DOM-E5141 - Intelligent Computational Media D (Perttu Hämäläinen)

- If you are interested in diving deeper into generative approaches, especially involving machine learning algorithms.
- Practical lecture on the algorithmic generation of video game content, computational music, sound installations, automatic testing and balancing of games, and intelligent image and 3D content.
- Prerequisites: experience in programming (e.g., Python, Javascript, Unity C#); high-school math.
- Further info on MyCourses and GitHub: https://github.com/PerttuHamalainen/MediaAI



Images and resources

Riverbed: https://wallup.net/river-stones-mountain/

Library of Babel illustration: Erik Desmazieres, https://cpb-us-e1.wpmucdn.com/blogs.cornell.edu/dist/b/5811/files/2019/08/Library_of_Babel-Desmazieres.jpg

Lindenmeyer system: Mortimer von Chappuis, https://commons.wikimedia.org/wiki/File:Fractal_Farn.gif

Game of Life, Marin Vlastelica Pogančić, https://towardsdatascience.com/the-game-of-life-the-legacy-of-john-conway-a86237180a4e

The Painting Fool / You Can’t Know My Mind Exhibition: from paper Colton, Simon, and Dan Ventura. "You Can't Know my Mind: A Festival of Computational Creativity." ICCC. 2014.

Gloomy Sunday video: http://www.memo.tv/works/gloomy-sunday/

StyleGAN2: from paper Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T. (2020). Analyzing and improving the image quality of stylegan. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 8110-8119).

The Painting Fool Example Portraits: http://www.thepaintingfool.com/galleries/emotionally_aware/index.html

The guardian website: https://www.theguardian.com/uk

Creative Adversarial Networks: from paper Elgammal, Ahmed, et al. "CAN: Creative adversarial networks generating “Art” by learning about styles and deviating from style norms." Proc. International Conference on Computational Creativity, ICCC 2017.

Dwarf Fortress: https://www.ign.com/articles/2019/03/16/dwarf-fortress-steam-story

Images and resources

ANGELINA: from Cook, M., Colton, S., & Gow, J. (2016b). The angelina videogame design system—part ii. IEEE Transactions on Computational Intelligence and AI in Games, 9(3), 254-266.

Sentient sketchbook: from Yannakakis, G. N., Liapis, A., & Alexopoulos, C. (2014). Mixed-initiative co-creativity. Proc. Foundations of Digital Games Conference.

NASA ST5 Spacecraft Antenna: https://en.wikipedia.org/wiki/Evolved_antenna#/media/File:St_5-xband-antenna.jpg

References

Boden, M.A.,(1990) The Creative Mind, Abacus, London

Bown, O. (2012). Generative and Adaptive Creativity: A Unified Approach to Creativity in Nature, Humans and Machines. Computers and Creativity (pp. 361–381). Springer.

Colton, S. (2008). Creativity Versus the Perception of Creativity in Computational Systems. Proc. AAAI Spring Symposium: Creative Intelligent Systems, 1–7.

Colton, S., Halskov, J., Ventura, D., Gouldstone, I., Cook, M., & Ferrer, B. P. (2015, June). The Painting Fool Sees! New Projects with the Automated Painter. In Proc. Int. Conference on Computational Creativity (ICCC), pp. 189-196.

Cook, M., Colton, S., & Gow, J. (2016a). The angelina videogame design system—part i. IEEE Transactions on Computational Intelligence and AI in Games, 9(2), 192-203.

Cook, M., Colton, S., & Gow, J. (2016b). The angelina videogame design system—part ii. IEEE Transactions on Computational Intelligence and AI in Games, 9(3), 254-266.

Elgammal, Ahmed, et al. "CAN: Creative adversarial networks generating “Art” by learning about styles and deviating from style norms." Proc. International Conference on Computational Creativity, ICCC 2017.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial networks. arXiv preprint arXiv:1406.2661.

Guckelsberger, C., Salge, C., Gow, J., & Cairns, P. (2017). Predicting Player Experience without the Player. An Exploratory Study. In Proceedings of the Annual Symposium on Computer-Human Interaction in Play (pp. 305-315).

Holland, J. H. (1992). Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press.

Holland, J. H. (1992). Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press.

References

Hornby, G., Globus, A., Linden, D., & Lohn, J. (2006). Automated antenna design with evolutionary algorithms. In Space 2006 (p. 7242).

Jennings, K. E. (2010). Developing Creativity: Artificial Barriers in Artificial Intelligence. Minds and Machines, 20(4), 489–501.

Jordanous, A. (2012). A Standardised Procedure for Evaluating Creative Systems: Computational Creativity Evaluation Based on What It Is to Be Creative. Cognitive Computation, 4(3), 246–279.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T. (2020). Analyzing and improving the image quality of stylegan. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 8110-8119).

Linkola, S., Guckelsberger, C., & Kantosalo, A. (2020). Action Selection in the Creative Systems Framework. In Proc. International Conference on Computational Creativity. The Association for Computational Creativity.

Runco, M. A. & Jaeger, G. J. (2012). The Standard Definition of Creativity. Creativity Research Journal, 24(1), 92–96.

Russell, S., & Norvig, P. (2003). Artificial intelligence: a modern approach. 2nd ed.

Saunders, R. (2009). Supporting Creativity Using Curious Agents. Proc. Workshop on Computational Creativity Support at the Conference on Human Factors in Computing Systems (CHI), 1–4.

Still, A. & d’Inverno, M. (2016). A History of Creativity for Future AI Research. Proc. Int. Conference on Computational Creativity (ICCC), 147–154.

Togelius, J., Yannakakis, G. N., Stanley, K. O., & Browne, C. (2011). Search-based procedural content generation: A taxonomy and survey. IEEE Transactions on Computational Intelligence and AI in Games, 3(3), 172-186.

Wiggins, G. A. (2006a). Searching for computational creativity. *New Generation Computing*, *24*(3), 209-222.

Wiggins, G. A. (2006b). A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems*, *19*(7), 449-458.

Yannakakis, G. N., Liapis, A., & Alexopoulos, C. (2014). Mixed-initiative co-creativity. Proc. Foundations of Digital Games Conference.