Procedural Content Generation Introduction

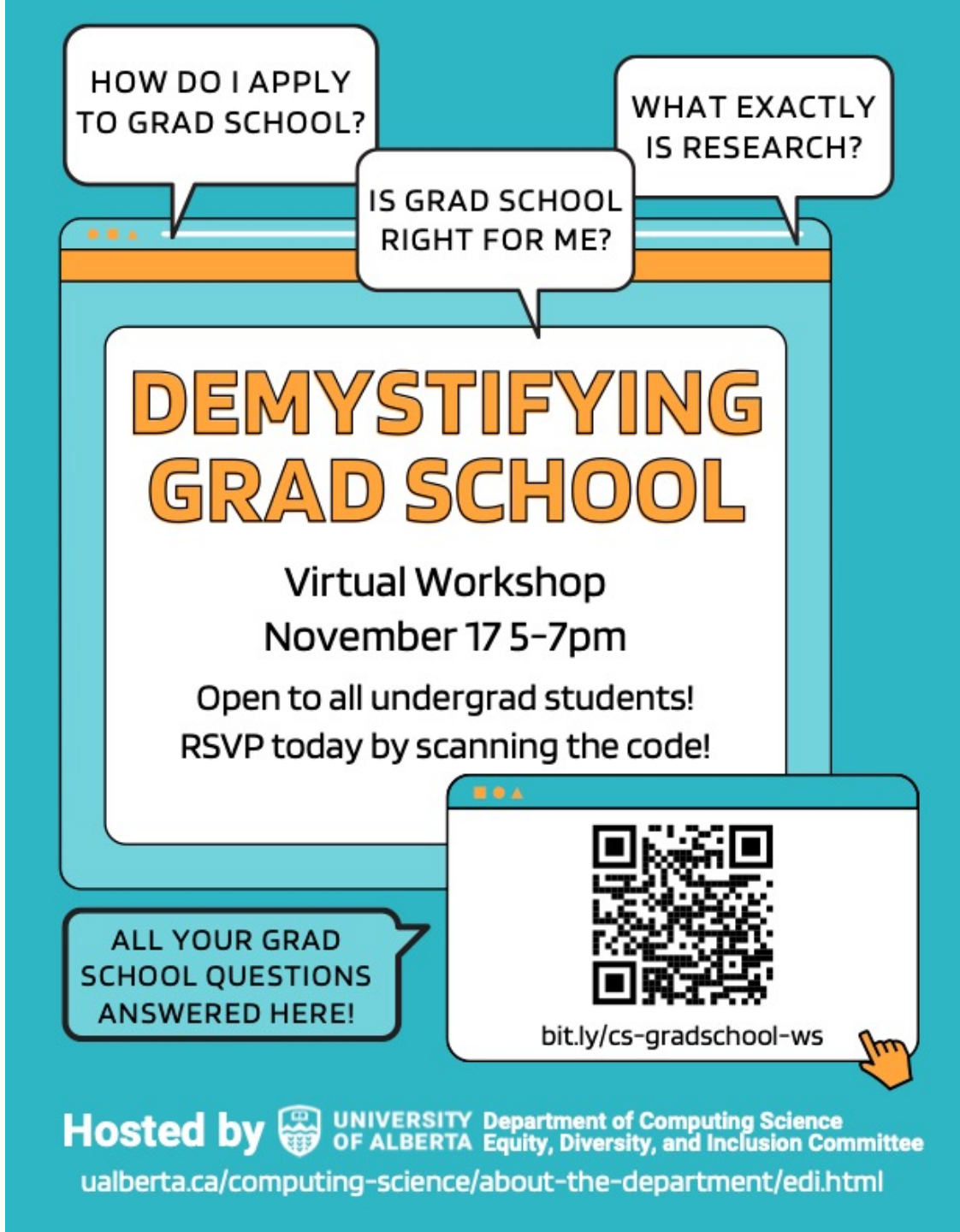
Matthew Guzdial

guzdial@ualberta.ca

Announcements

- Quiz 4 released tomorrow at 11am
- Tomorrow online help session 5-8pm
- No lecture Friday! (but more online help from 11am-1pm)
- Quiz 4 due Sunday at 11am
- Assignment 4 due after reading week

Demystifying Grad School
Nov 17 5-7pm
bit.ly/cs-gradschool-ws



Future of Game AI voting (PQ1)
https://forms.gle/Zqzswt31hNyvEoVG8
https://tinyurl.com/guz-pq26a

- Reinforcement Learning in Games, Automated Playtesting, Game AI in Academia, AI for Automated Game Playing, Balancing Game AI, "How AI can change cinematic cutscenes", Generated dialogue and story
- AI-based Game Design
- AI for Game Design
- Procedural Content Generation via Machine Learning
- Mixed-initiative PCG (human + AI designing)
- Further discussion on covered topic

Procedural Content Generation

- Computationally representing a design process.
- Large time and expertise requirements.
- Used to create new content for a single game.

Procedural Content Generation (PCG)

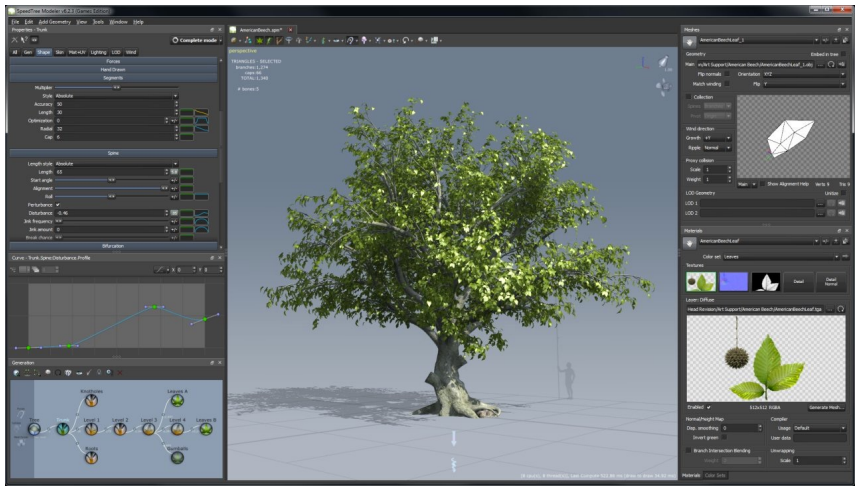
- Algorithmic generation of content
  - Game bits
  - Game space
  - Game scenarios

- Stolen terms: Procedural Content Generation for Games: A Survey
  - https://course.ccs.neu.edu/cs5150f13/readings/hendrikx_pcgg.pdf

- Recommended reading: Chapter 4 of Yannakakis and Togelius “Artificial Intelligence and Games” http://gameaibook.org/book.pdf
- Shout out to Adeel Zafar.

Game Bits

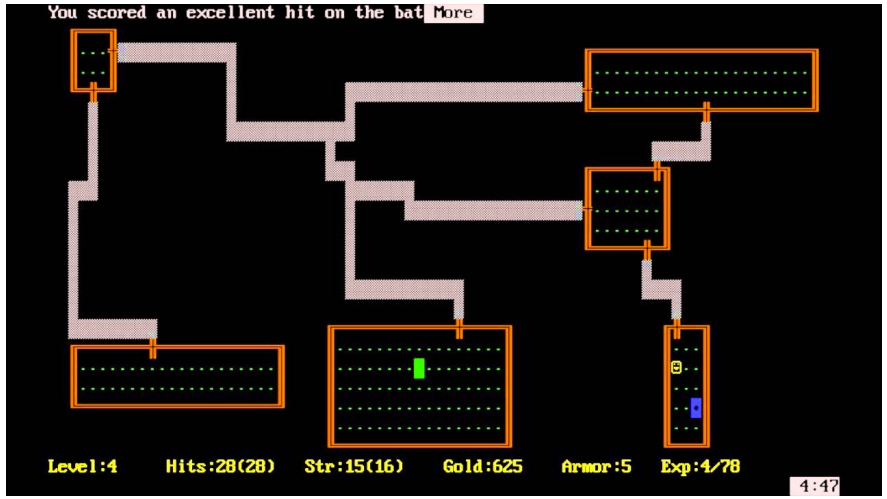
Borderlands series



SpeedTree

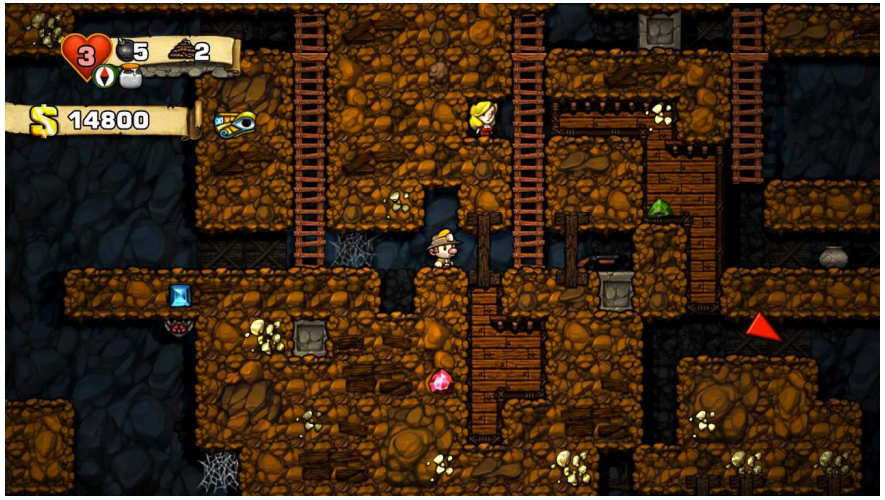
No Man’s Sky

Game Space



Rogue (1980)

Remnant, From the Ashes



Spelunky

Minecraft

Game Scenarios

Skyrim, radiant quests

Dwarf Fortress

Elsinore

Caves of Qud

TLDR PCG isn’t Magic

Imagine trying to write down instructions so someone you will never meet can create a level/puzzle/creature/etc. for your video game.

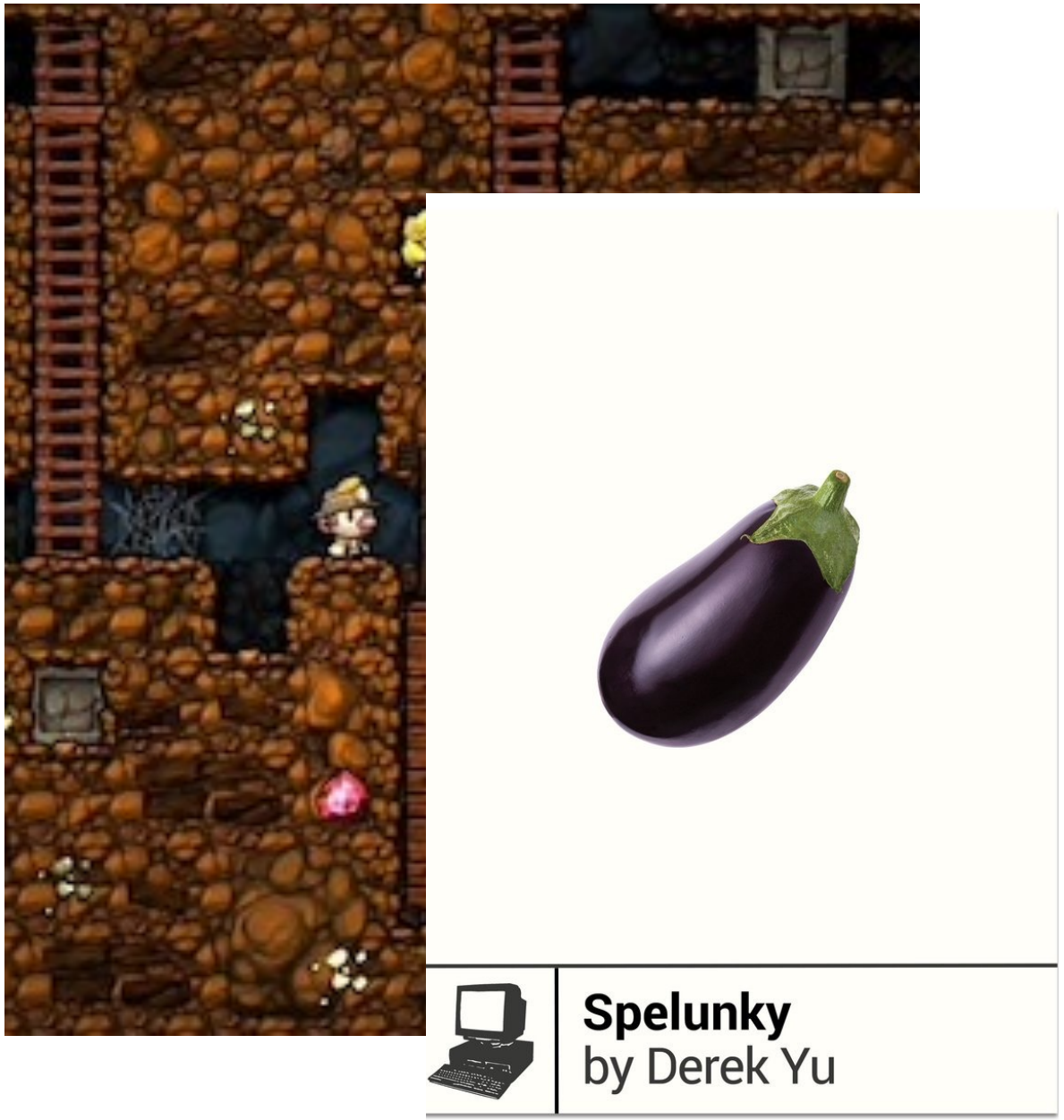
Case Study: Mass Effect Andromeda

- Mass Effect Andromeda in development for 5 years
- Majority of development occurred in final 18 months
- The culprit? PCG

https://kotaku.com/the-story-behind-mass-effect-andromedas-troubled-five-1795886428

Case Study: Spelunky

- Developed in large part by a single-person, Derek Yu.
- PCG uses a constructive grammar of hand-authored chunks.
- PCG makes for a larger game than a single-developer could create.



Three Major Method Categories

1. **Constructive Methods**: Methods that build up content piece-by-piece according to particular rules, where the pieces are either hand-authored or generated.
2. **Search-based Methods**: Define a search space where every point is a usable piece of content, then optimize this space according to some hand-authored heuristics and operators.
3. **Machine learning-based Methods**: Train on existing data and try to produce novel data that is similar but not identical to the original data. Not gonna talk about this today.

Constructive Methods

All constructive methods can be thought of as requiring building a **Grammar** (original concept from linguistics).

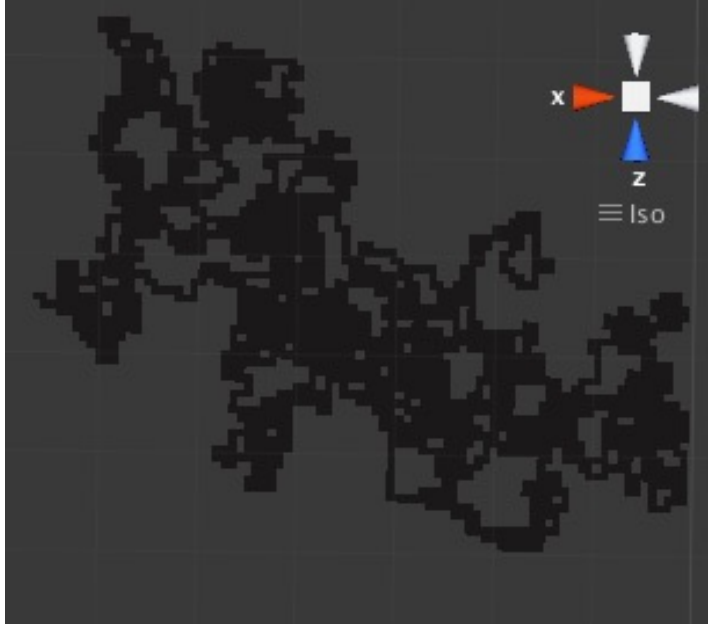
Grammar: Composed of two parts

- **Tokens**: The basic building blocks of the grammar. In language, these are words or phrases.
- **Rules**: How these tokens are allowed to be put together.

Quick Overview: 3 Noteworthy Methods

- Cellular Automata (early days pf PCG very popular)
- Generative Grammar (most popular current method)
- Constraint-satisfaction Problems (niche, but becoming less so)

Cellular Automata



Stanislaw Ulam and John von Neumann (1940)

Conway's Game of Life made famous (1970s)

Example for terrain:
https://www.youtube.com/watch?v=OumDj9PNWjE

At each position, make decision based on rules.

Local decisions percolate through the content until some hand-defined stopping point.

Generative Grammar

Well-suited for small stories, levels, creatures, etc.

Game Maker’s Toolkit
“How (and Why) Spelunky Makes its Own Levels | Game Maker's Toolkit”
https://www.youtube.com/watch?v=Uqk5Zf0tw3o

Typically built in a single-shot, no backtracking or altering after something is generated.

Time-consuming to create and takes PCG design expertise to get right.

Unity Example

PQ2: https://forms.gle/mGCumheLwEQnbDQ88
https://tinyurl.com/guz-pq26b

When do you think it’s appropriate to employ PCG in a game? (You’ve only “seen” one type so far, but hopefully you get the gist)

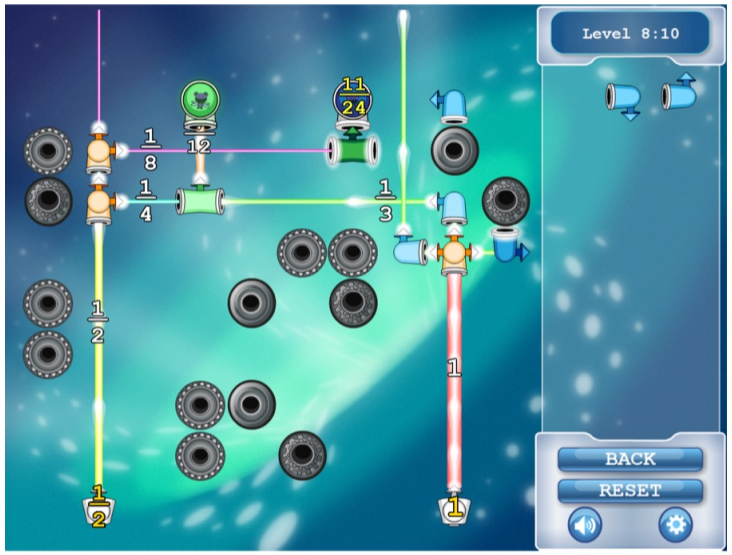
My Answer

- Definitely not “always”.
- When there’s an existing tool (e.g. SpeedTree) that you can just plug and play.
- When you are trying to create a game that would not otherwise be possible without PCG.
  - Even then, be aware of how much time it will take. PCG does not save time in developing a game.

Constraint-Satisfaction Problems (CSPs)

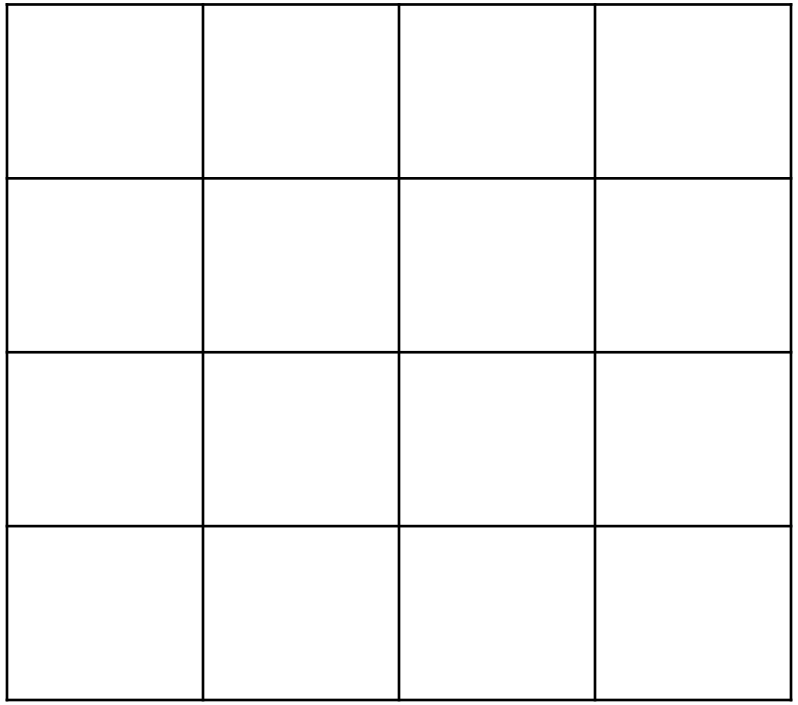
Well-suited for puzzle games or where precision is key.

Dense collection of rules, where each slot stores every possible solution until there's only one possible solution.

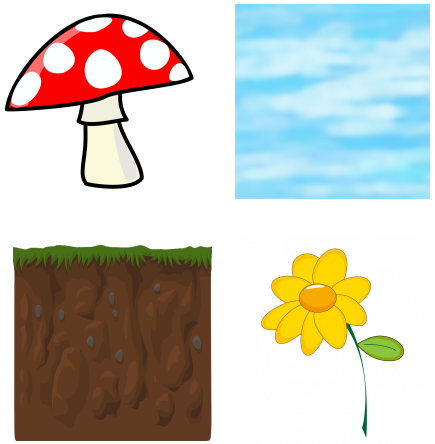


Refraction 2

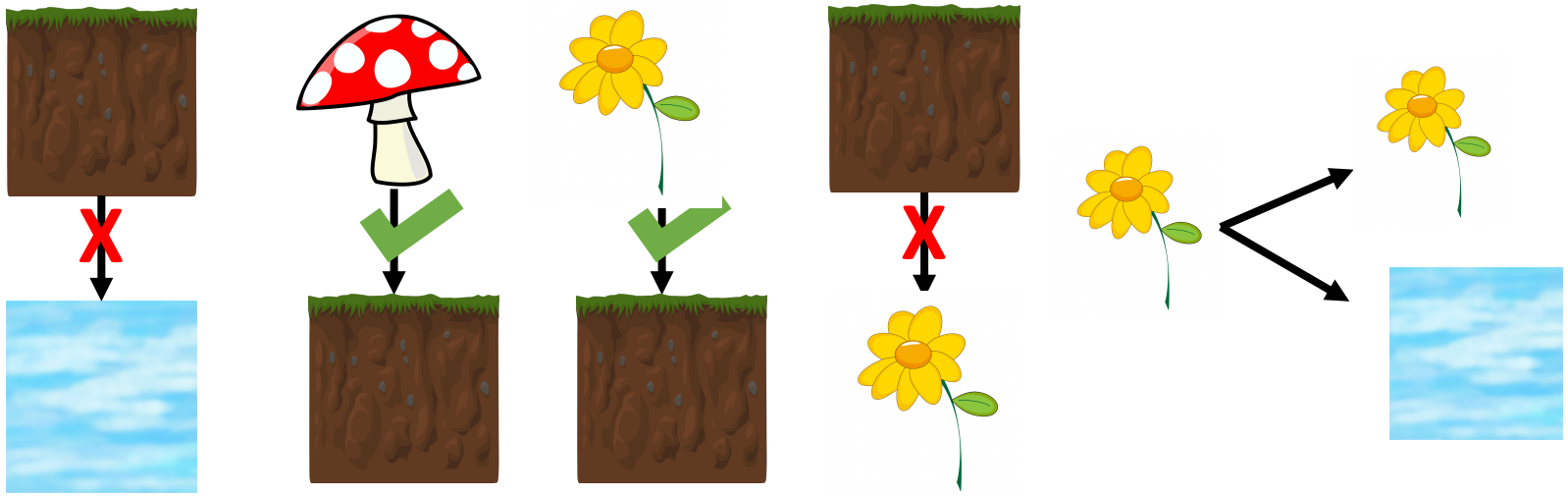
CSP Example



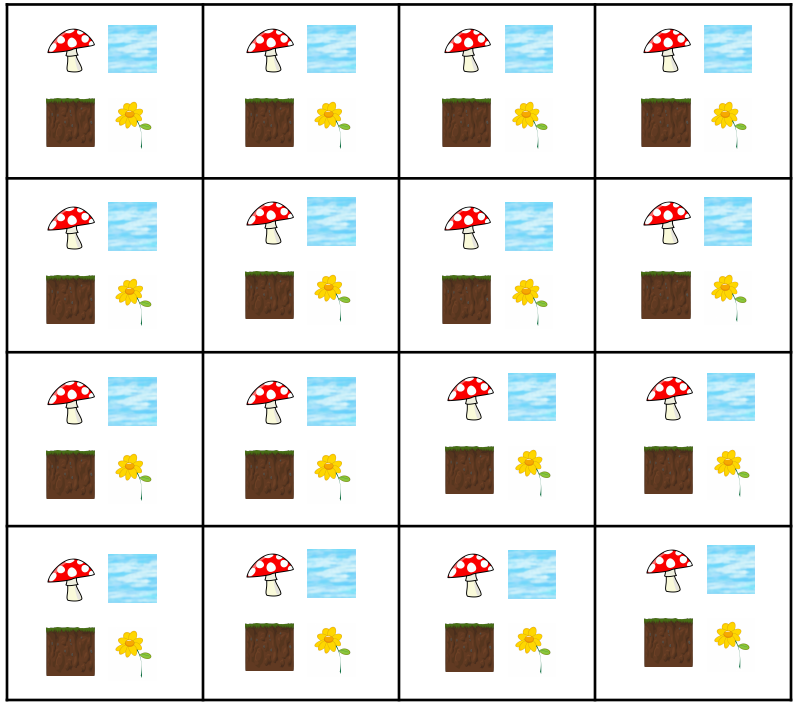
Tokens



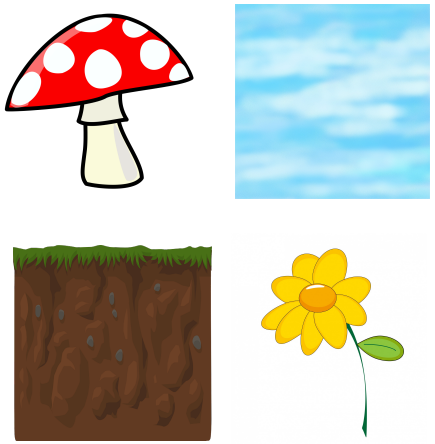
Rules



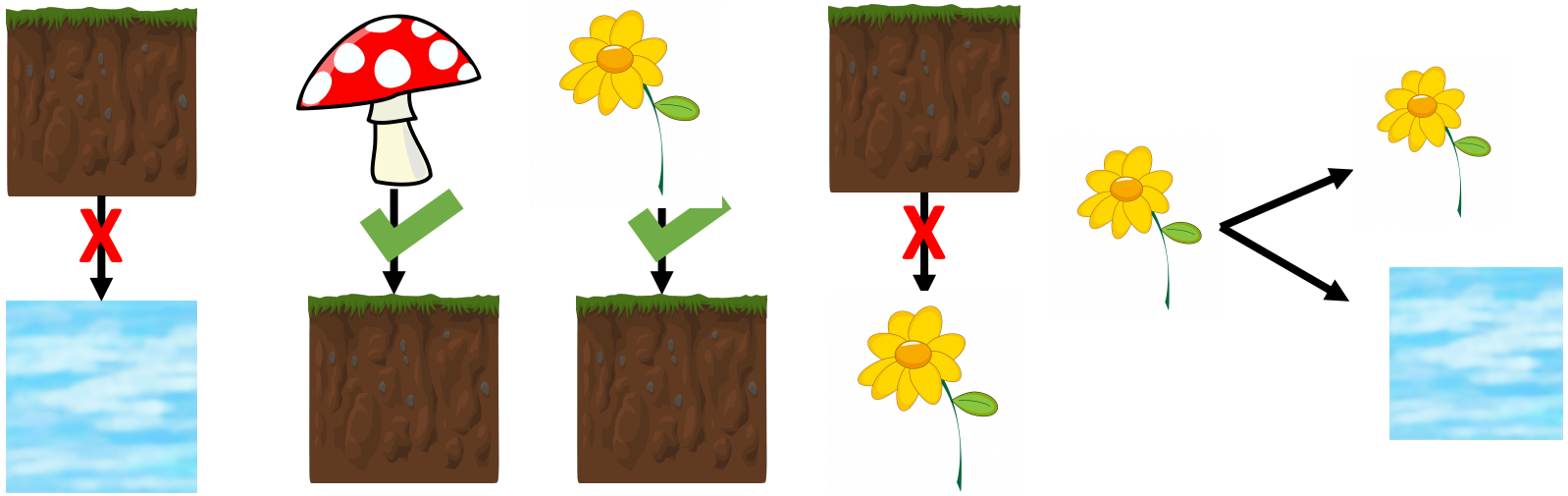
CSP Example Step 0



Tokens

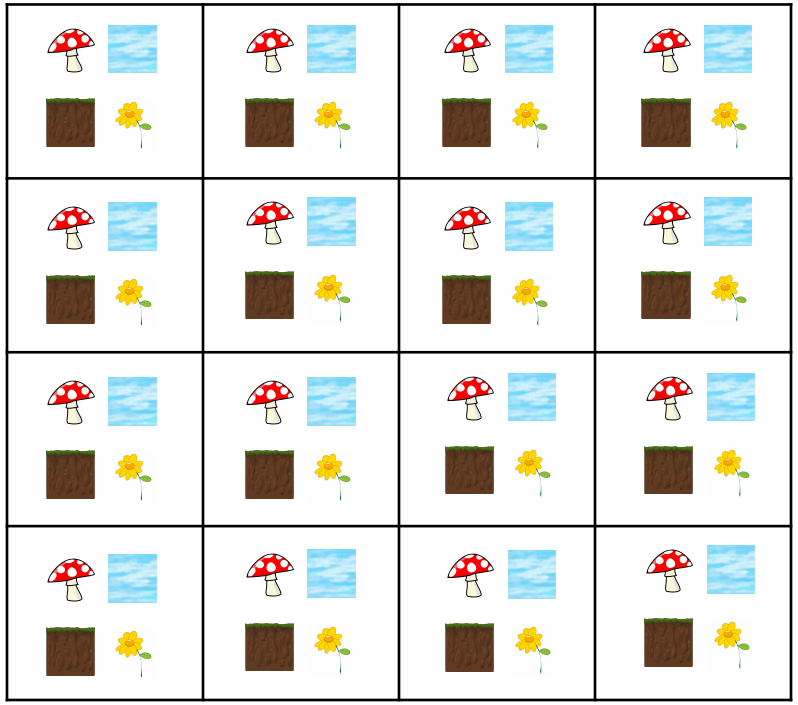


Rules

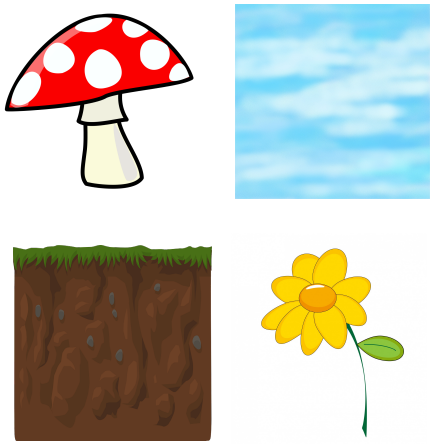


CSP Example Step 1

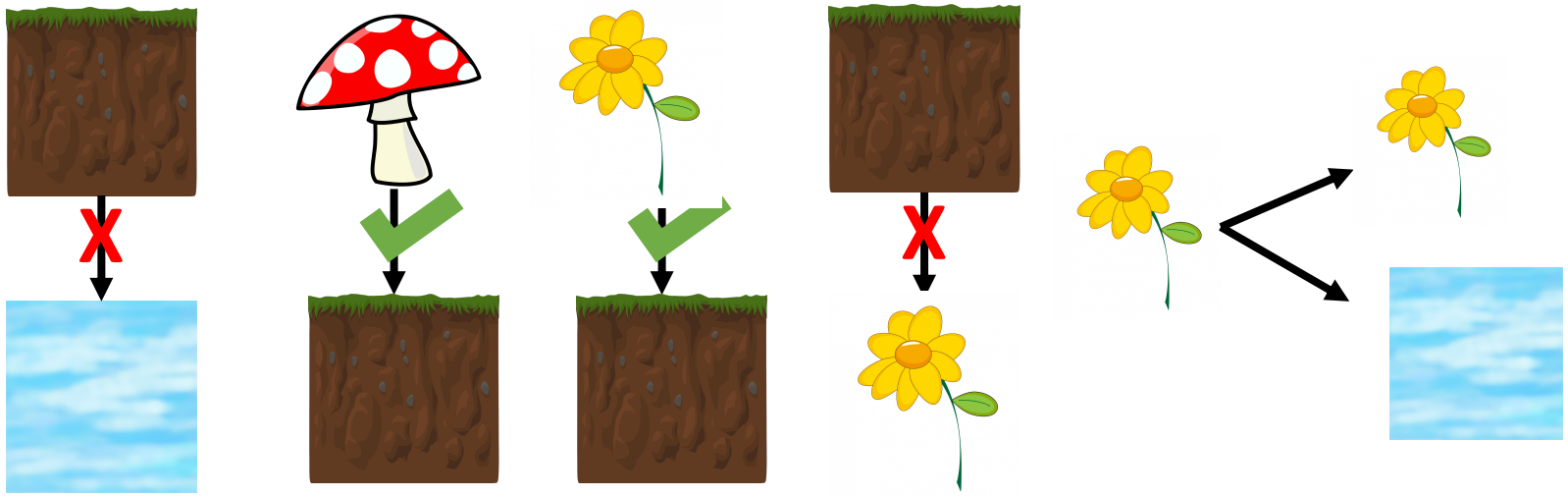
Decision



Tokens

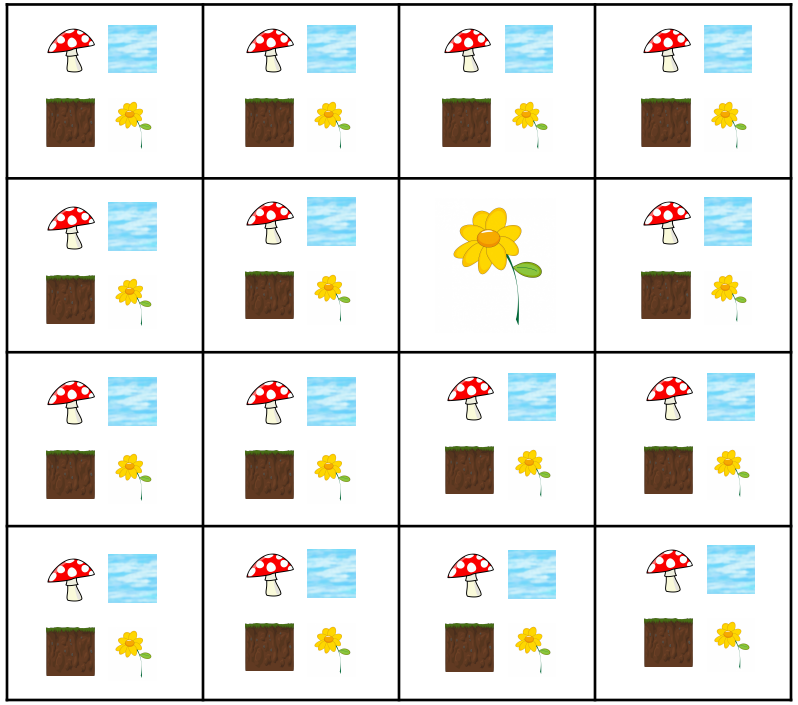


Rules

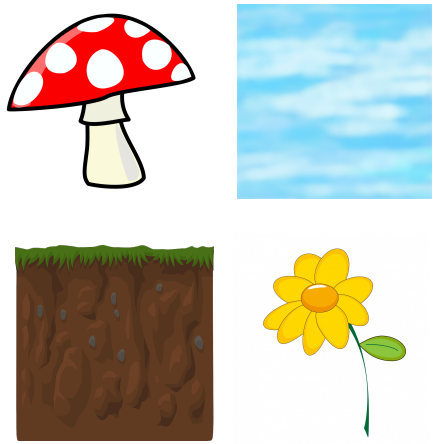


CSP Example Step 1

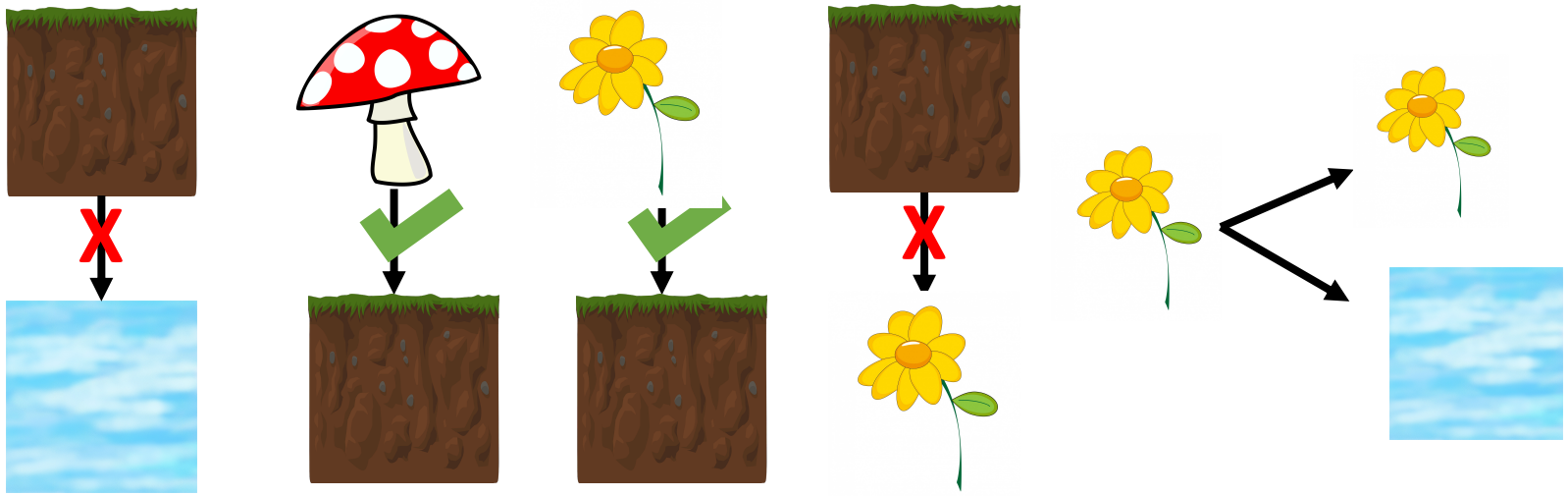
Decision



Tokens

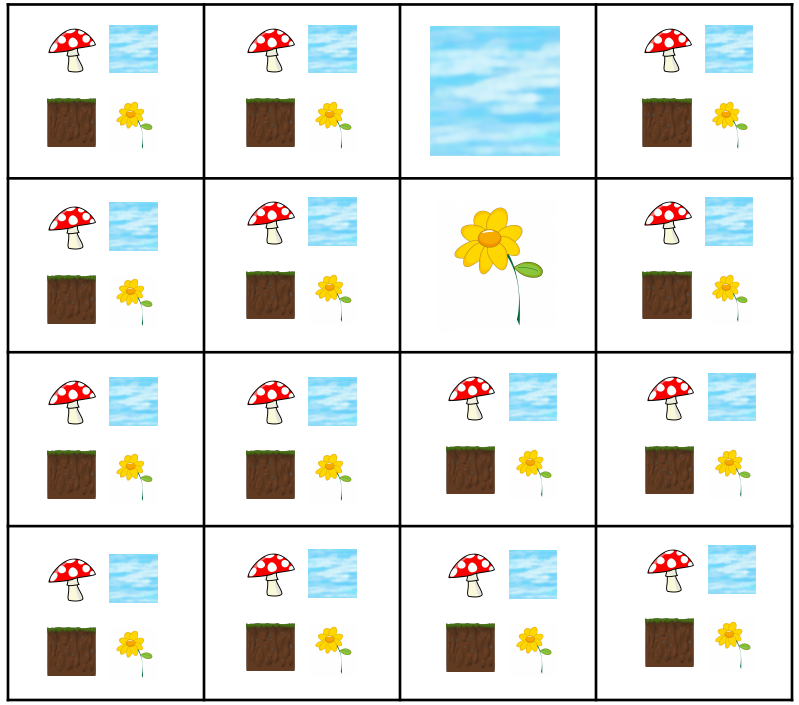


Rules

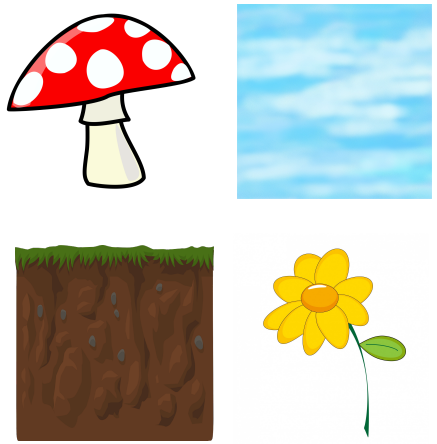


CSP Example
Step 1

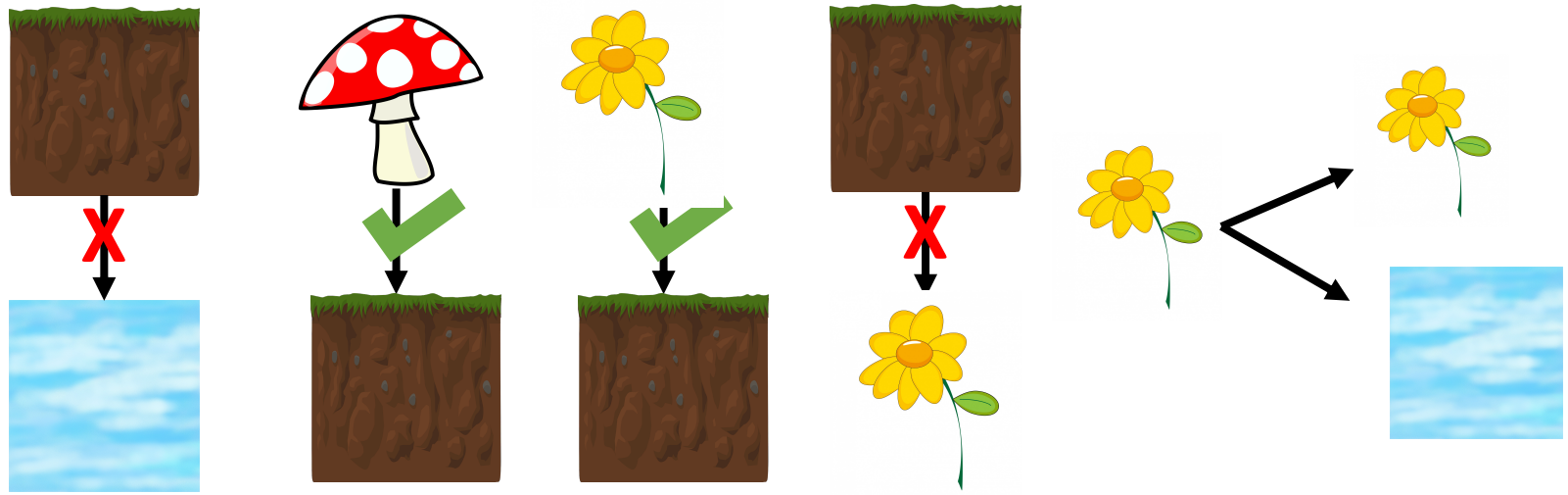
Implication



Tokens

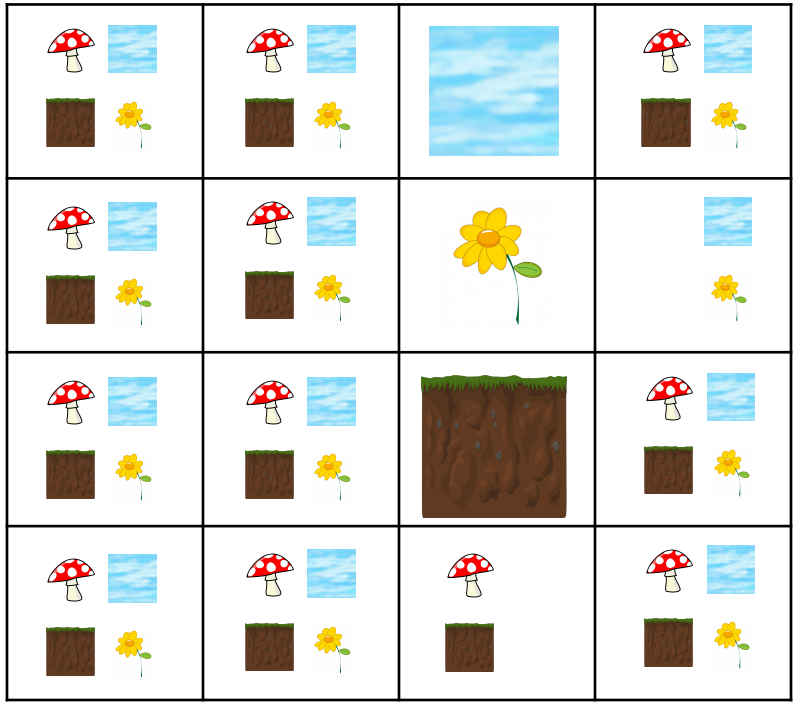


Rules

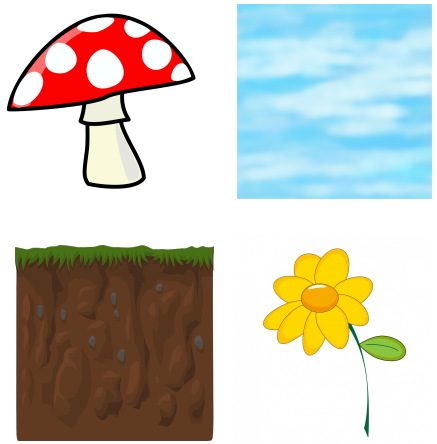


CSP Example Step 1

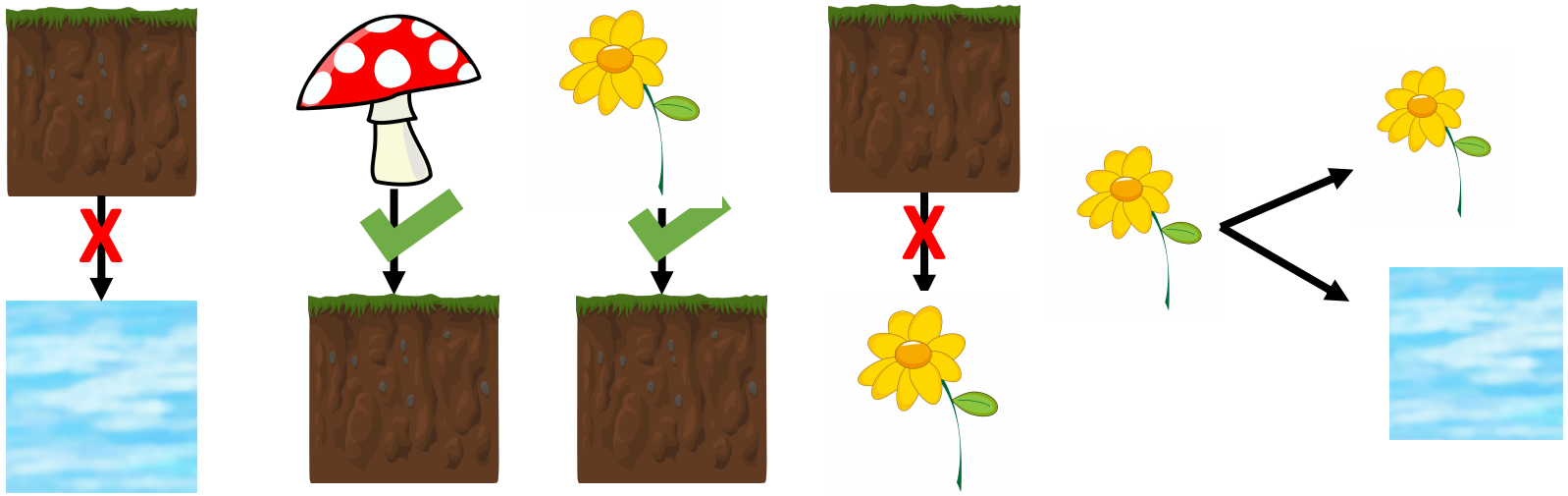
Implication



Tokens

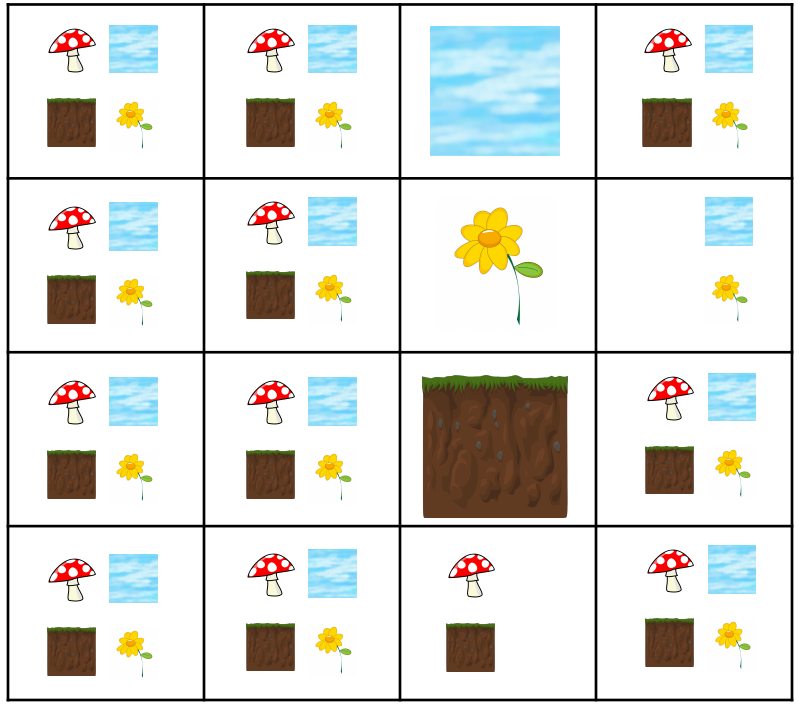


Rules

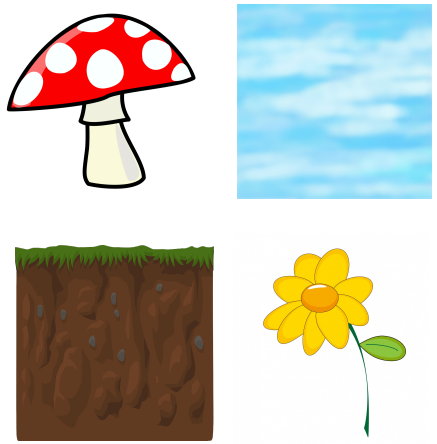


CSP Example Step 2

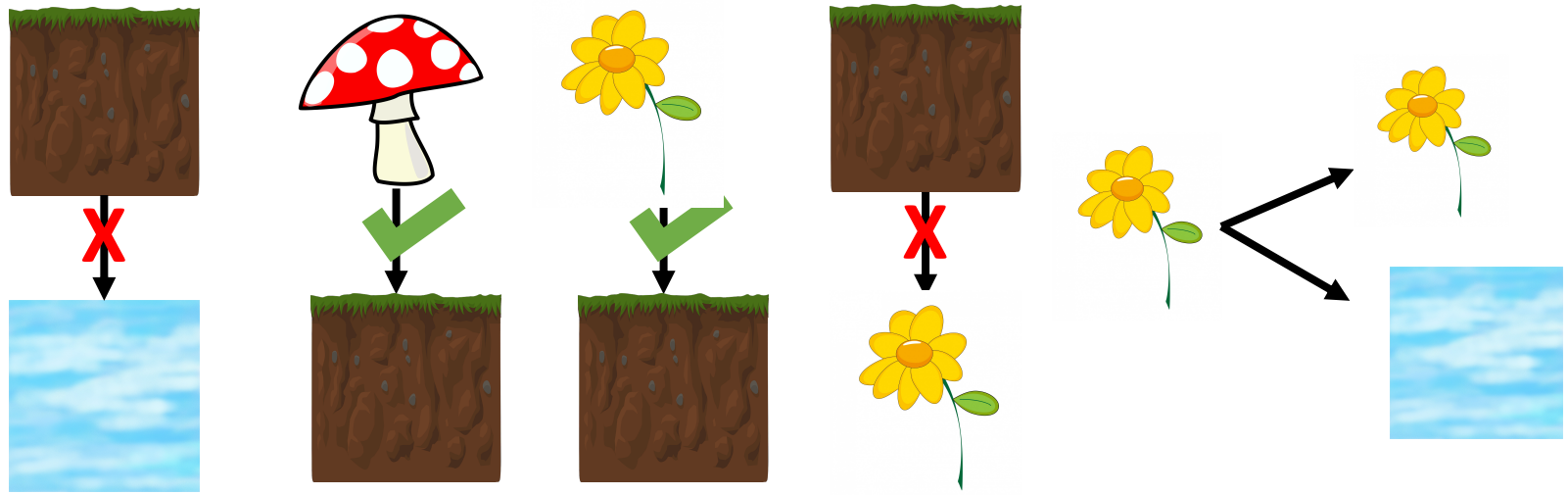
Decision



Tokens

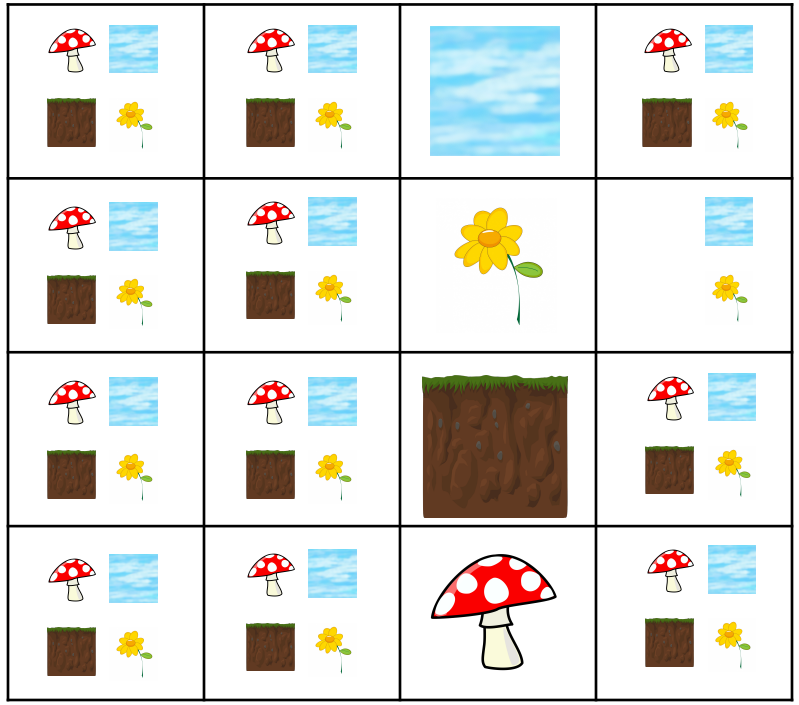


Rules

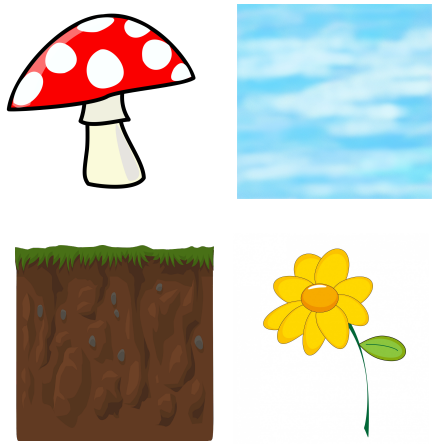


CSP Example Step 2

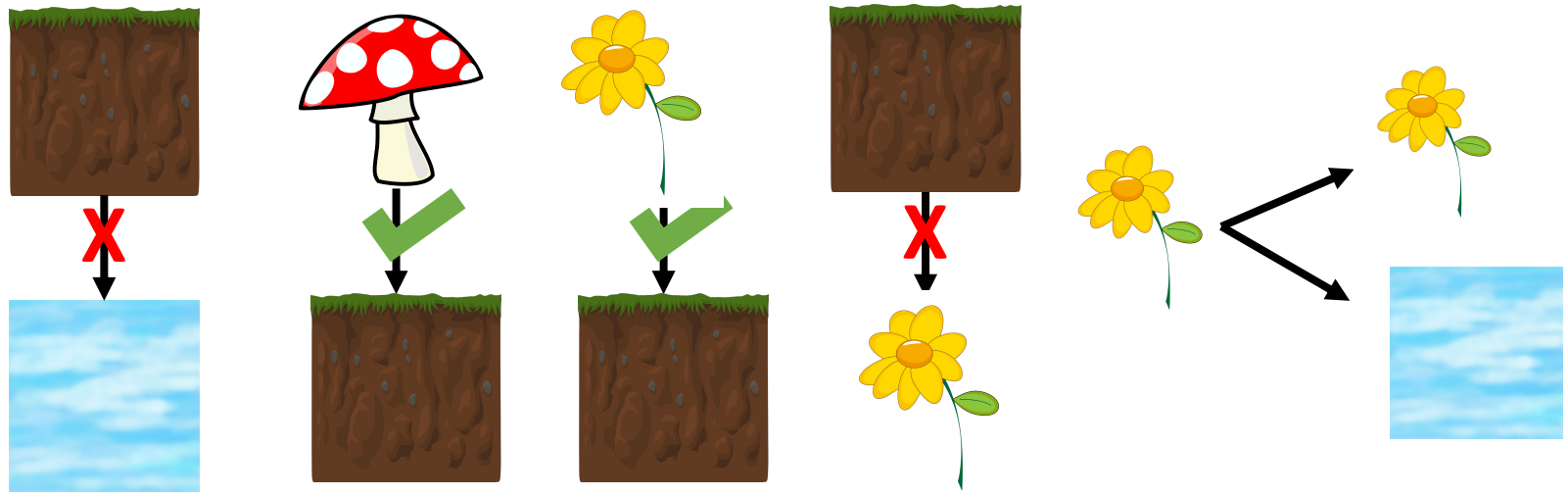
Decision



Tokens

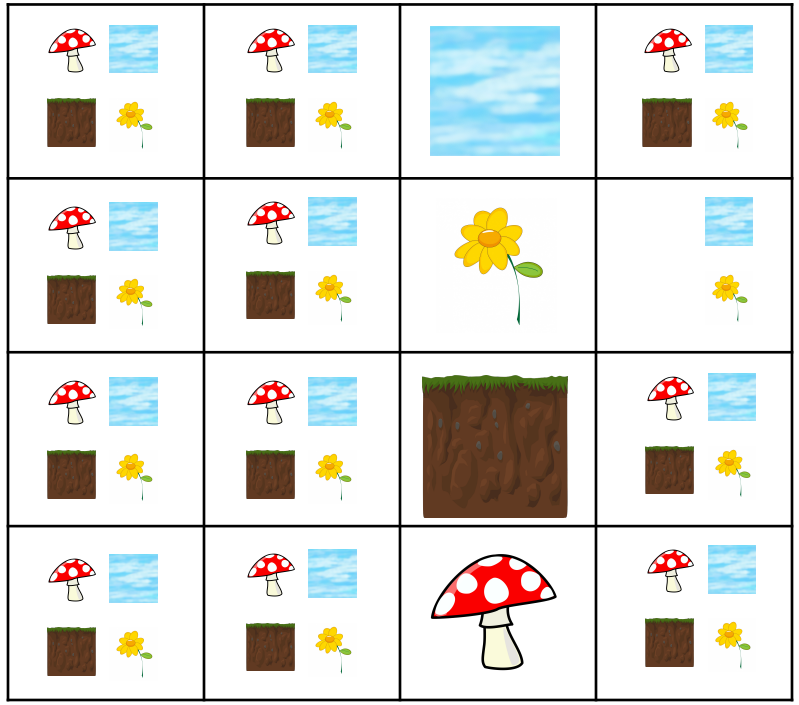


Rules

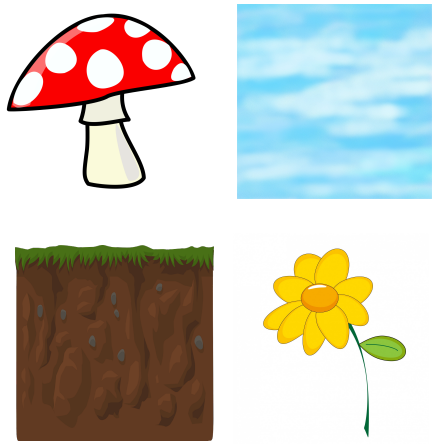


CSP Example Step 2

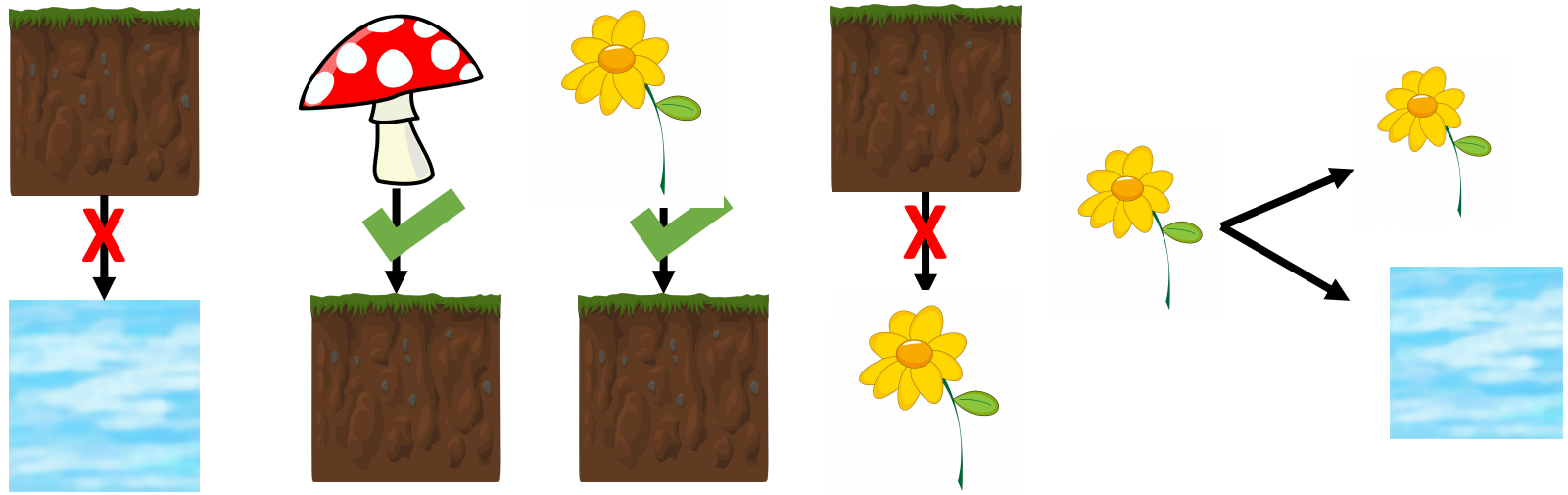
Implication



Tokens

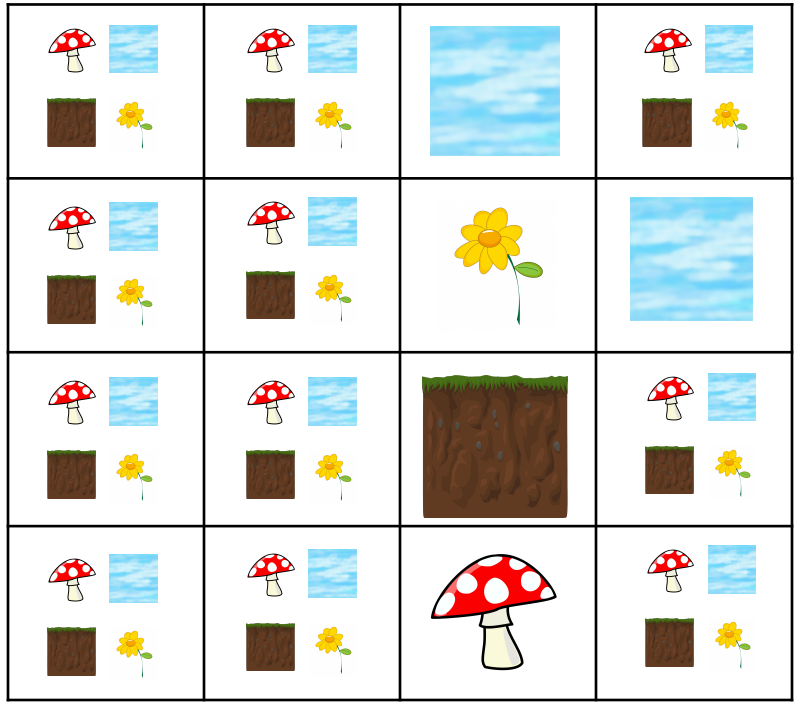


Rules

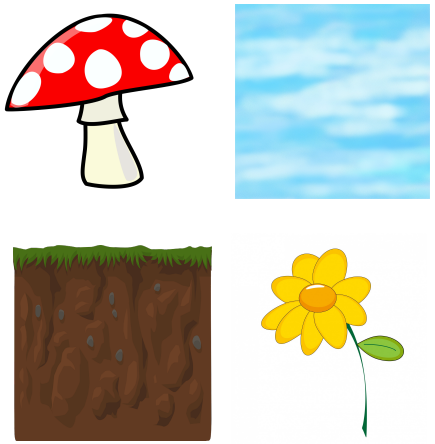


CSP Example
Step 3

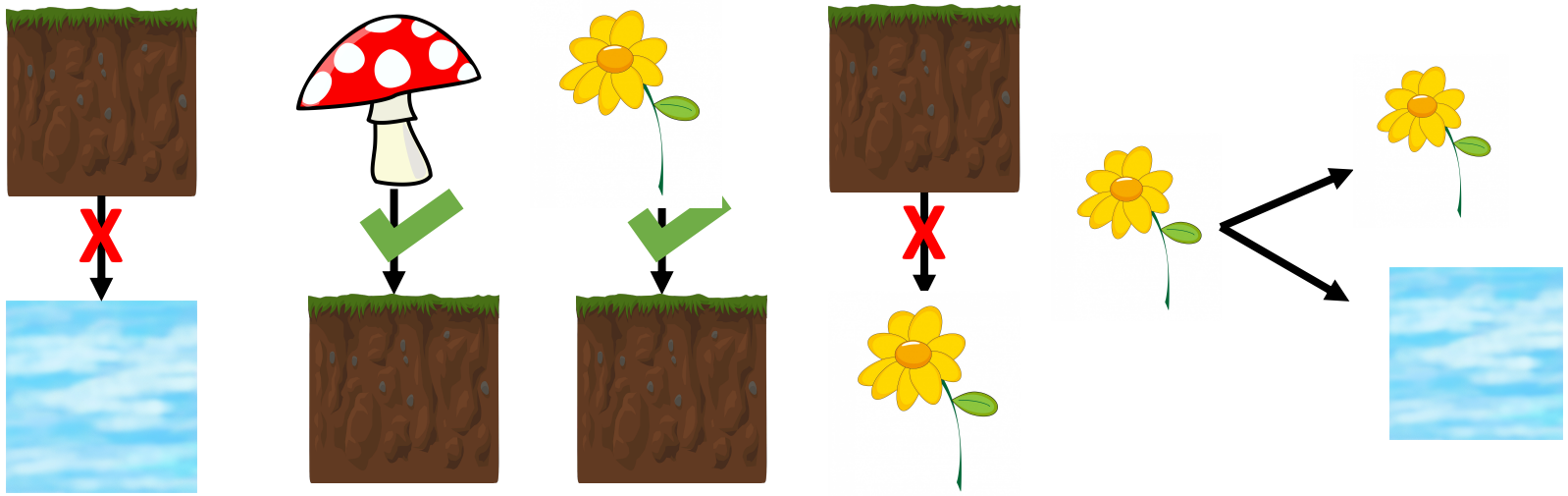
Decision



Tokens

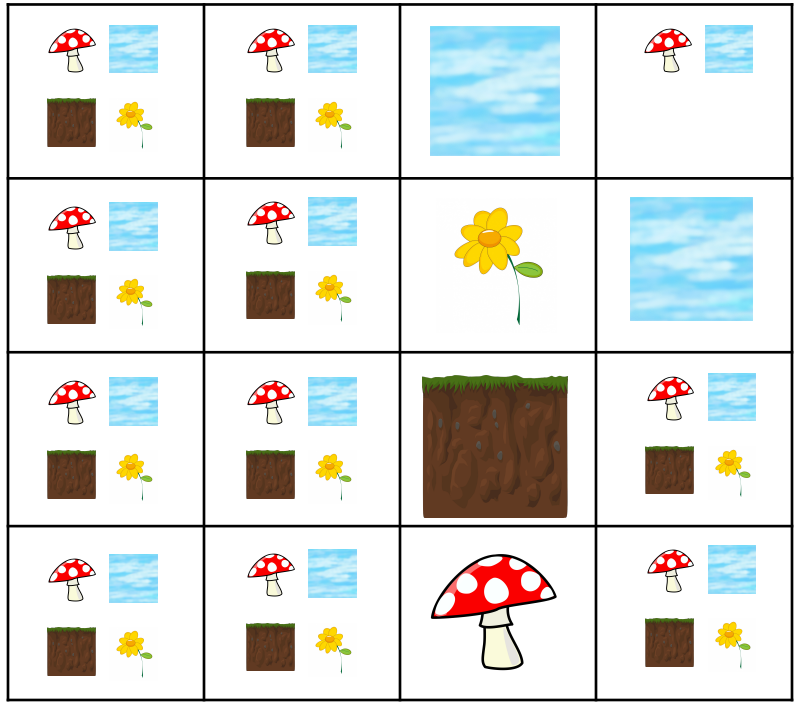


Rules

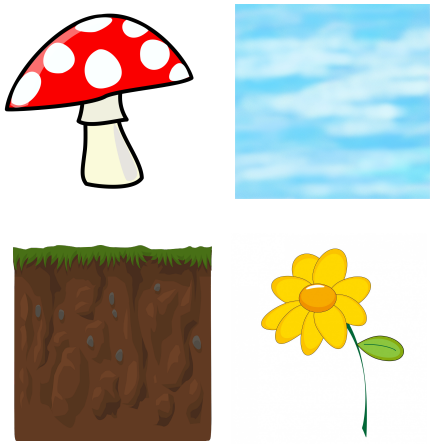


CSP Example Step 3

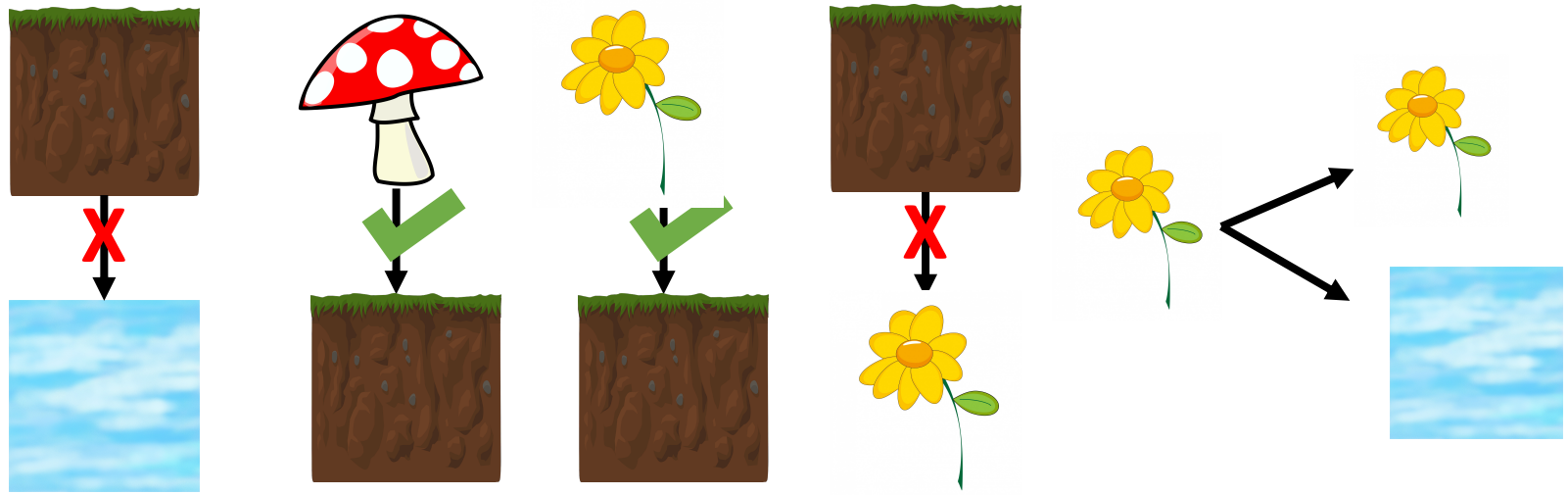
Implication



Tokens

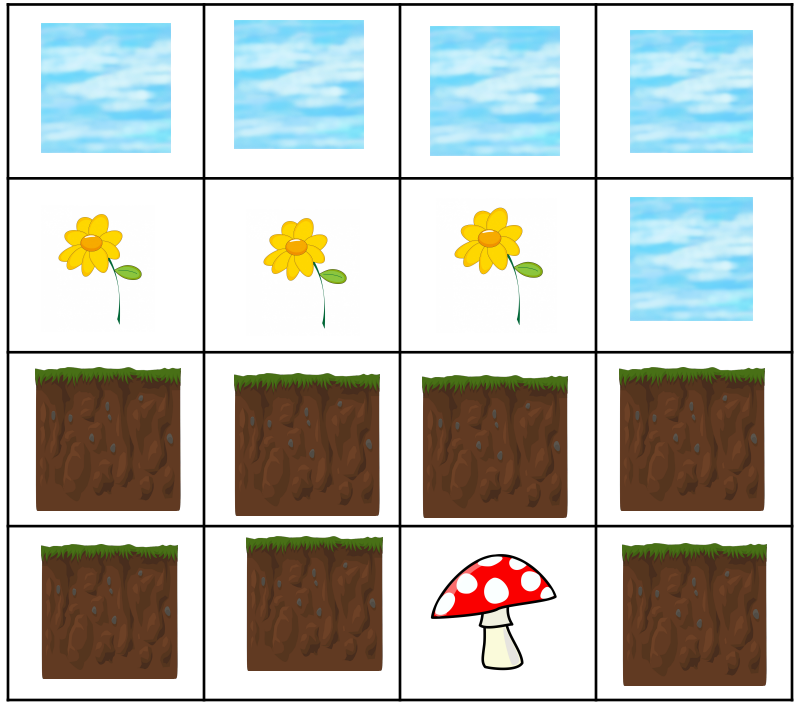


Rules

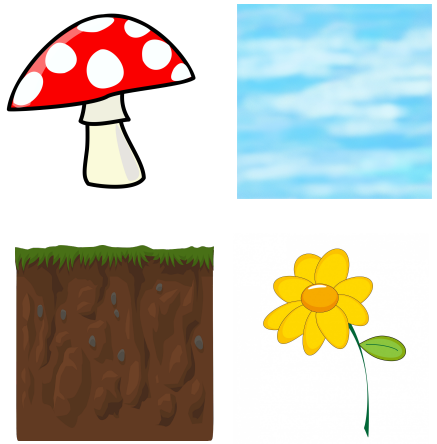


CSP Example
Step N

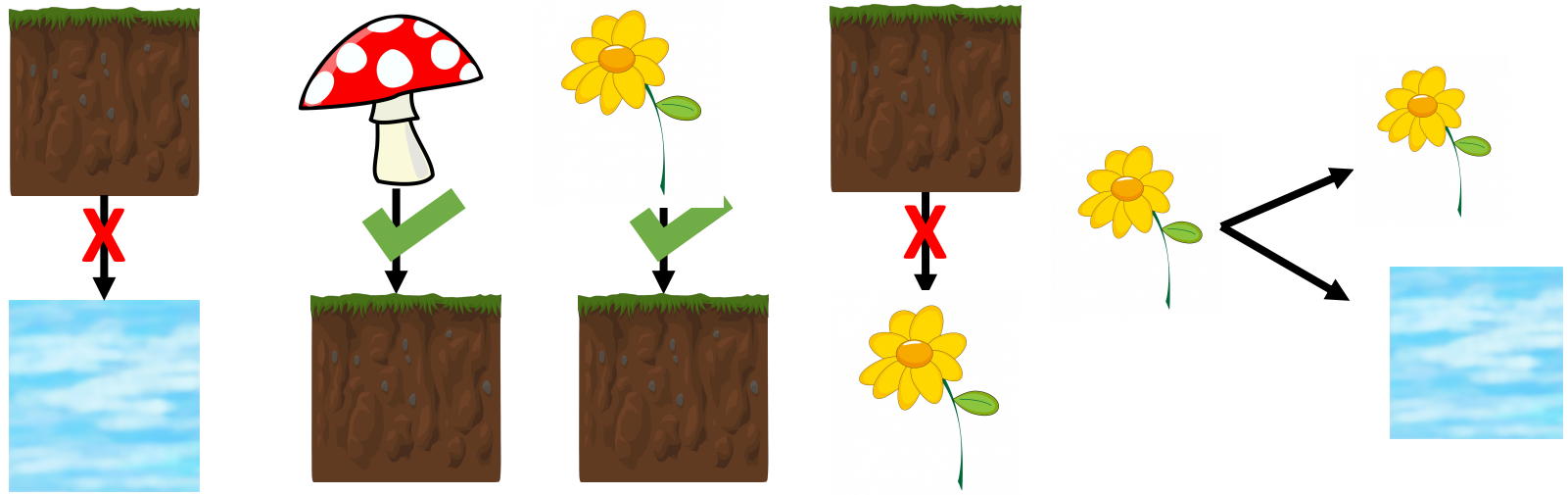
Option 1



Tokens

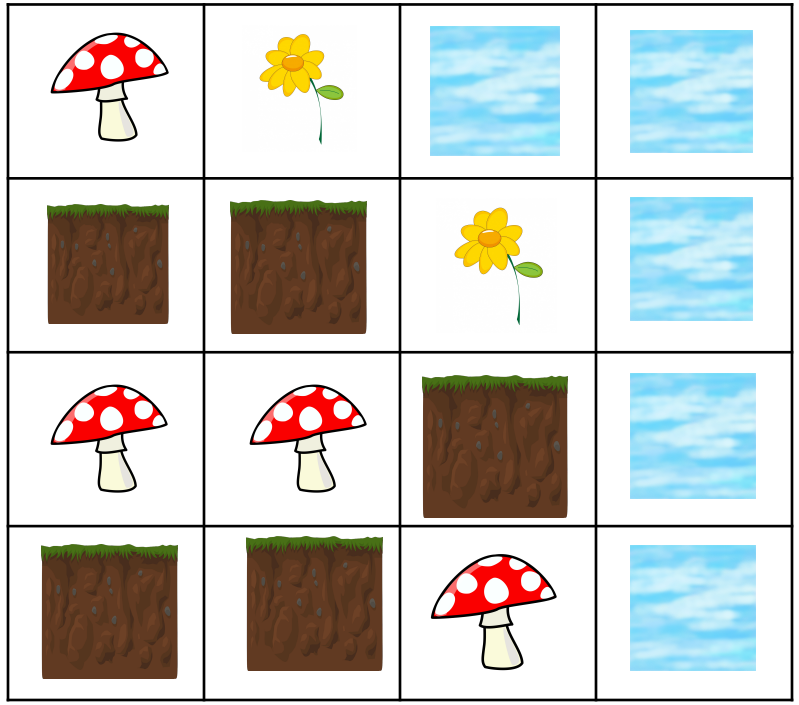


Rules

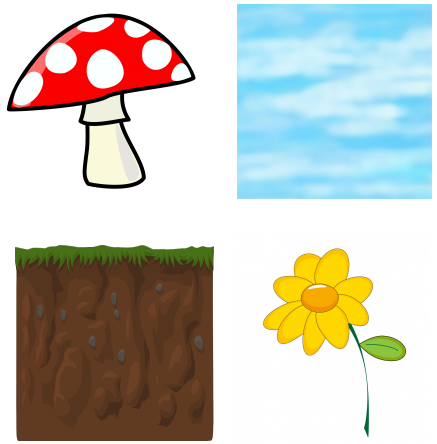


CSP Example
Step N

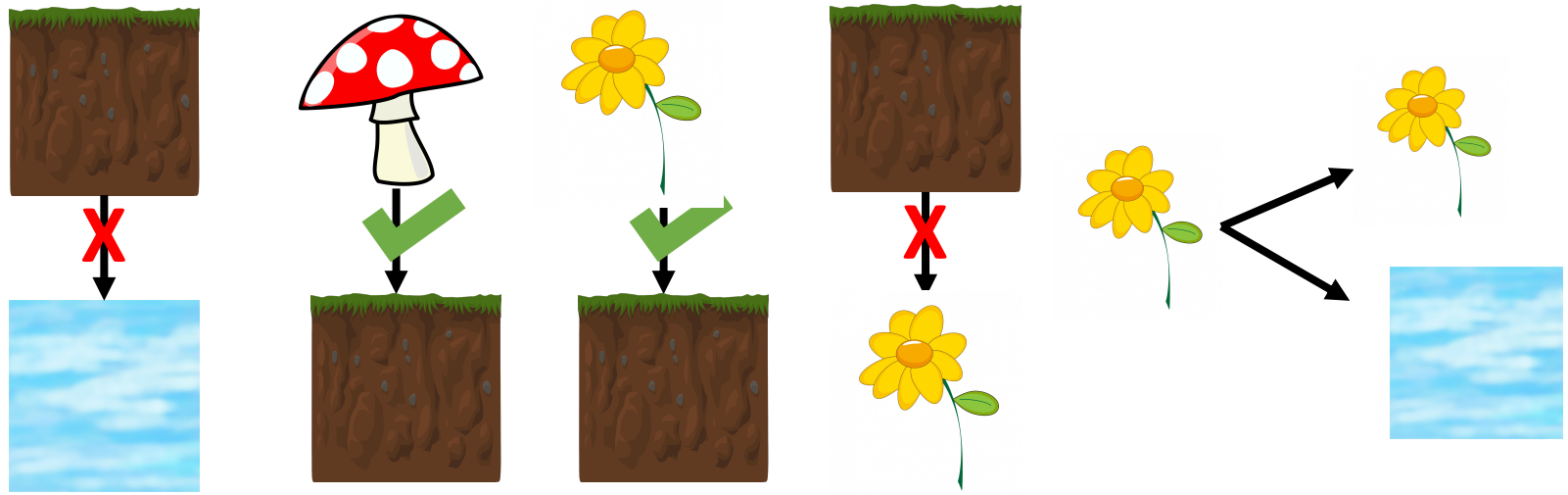
Option 2



Tokens



Rules



CSP Issues

- Hard to get a good set of rules!
  (over-engineering/emergence)
- Potentially slow (slower the more rules, backtracking).
- More on this after reading week

Search-based Methods

Define some space (*S*) where every element (*e*) in that space is a valid piece of content. Then plug in a standard AI search method to find particular content according to a heuristic/fitness function.

Fitness(*e*): Function that given *e* quantifies the quality of that content

Search-based Methods are generally not used in industry or indie game development.

Genetic Algorithms

- By far the most popular search-based method, also called Evolutionary Search.
- A popular search method in and out of games, inspired by Darwin's theory of evolution (loosely)
- More on this after reading week

Darwin's Demons

Petalz

General class question:

What makes a good measure of quality for game content?

My Answer: IDK

Search-based Methods Issues

- It’s really hard to determine a good measure of game content quality!
- Potentially very slow.
- Much more difficult to control than constructive methods.
- More on this after reading week

Much more out there

Chapter 4 of Yannakakis and Togelius “Artificial Intelligence and Games” http://gameaibook.org/book.pdf

AI and Games YouTube: https://www.youtube.com/user/tthompso

Conferences: FDG (PCG Workshop), AIIDE (EXAG workshop), AAAI, and IJCAI.

Quiz Review (time permitting)
Topics (be able to implement/walk through): Model-based player modelling, K means, K medoids, GMMs, Single-linkage, KNN, Linear/Logistic Regression, and Decision Trees

Topics (remember at high level): Spatial Representations, Path Planning, and Decision Making