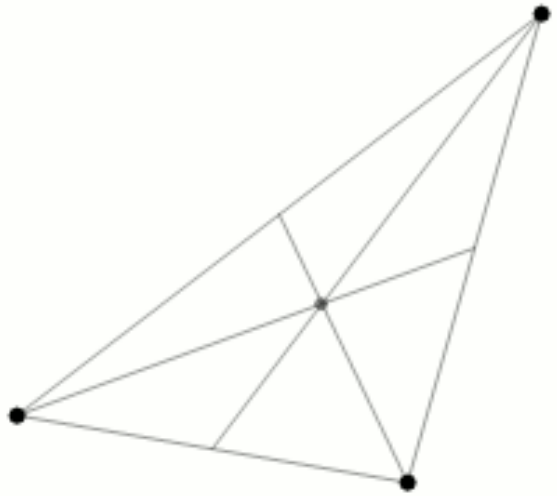
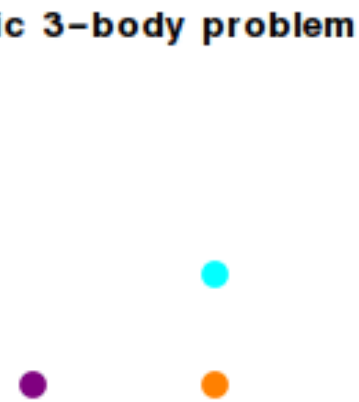
Cellular Automata

Is the unsolveability of F=ma something generally accepted, or is there debate over this?

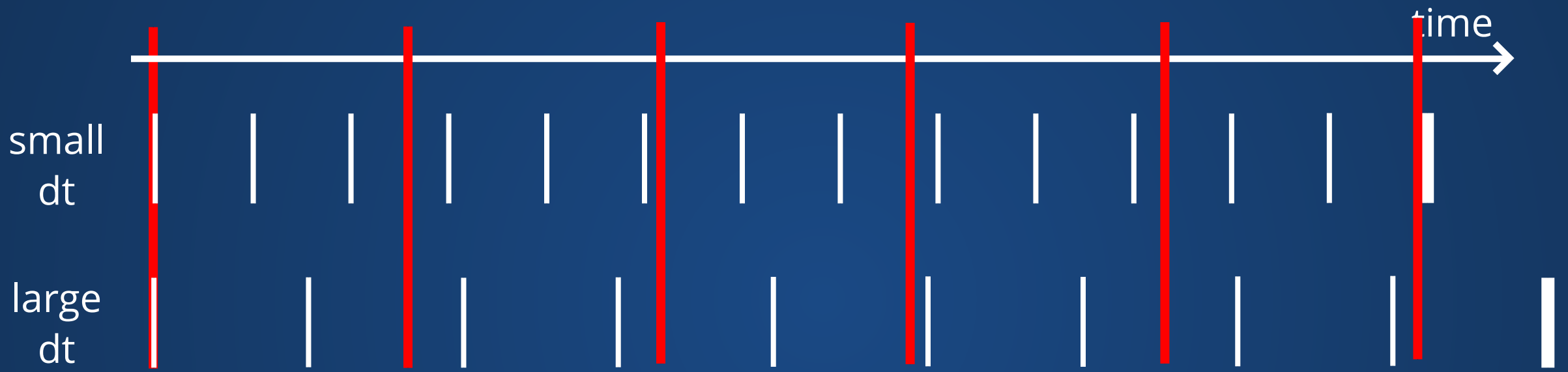
" There is no general closed-form solution to the three-body problem, meaning there is no general solution that can be expressed in terms of a finite number of standard mathematical operations.





Images from https://en.wikipedia.org/wiki/Three-body_problem

Does shrinking the timestep require processing to do more work vs a larger timestep?

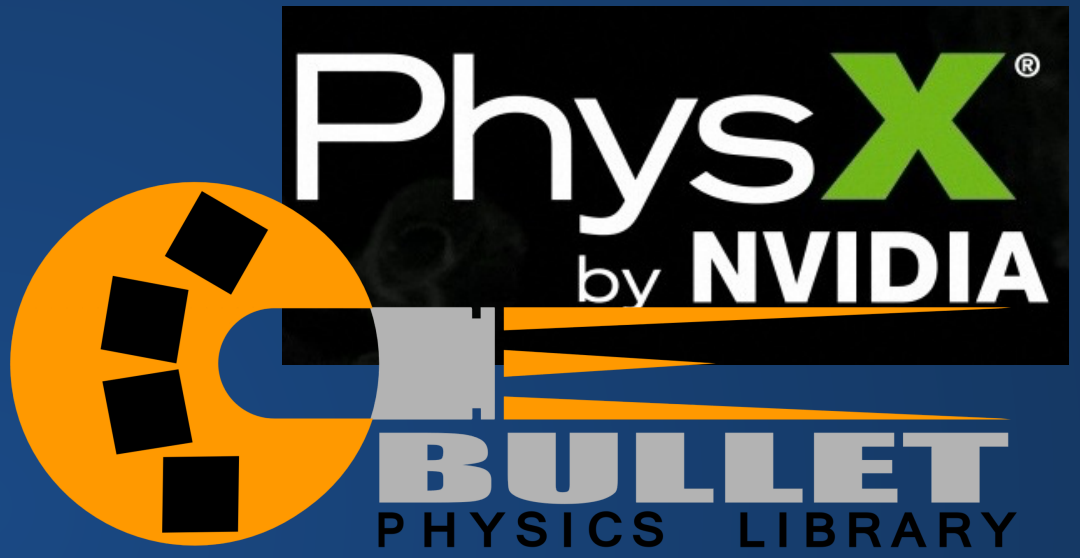


Computation for physics

Target draw time

Can all this physics stuff be done in Unreal? Or some other software?

Pretty much every modern game engine will have some sort of physics simulation code in it.

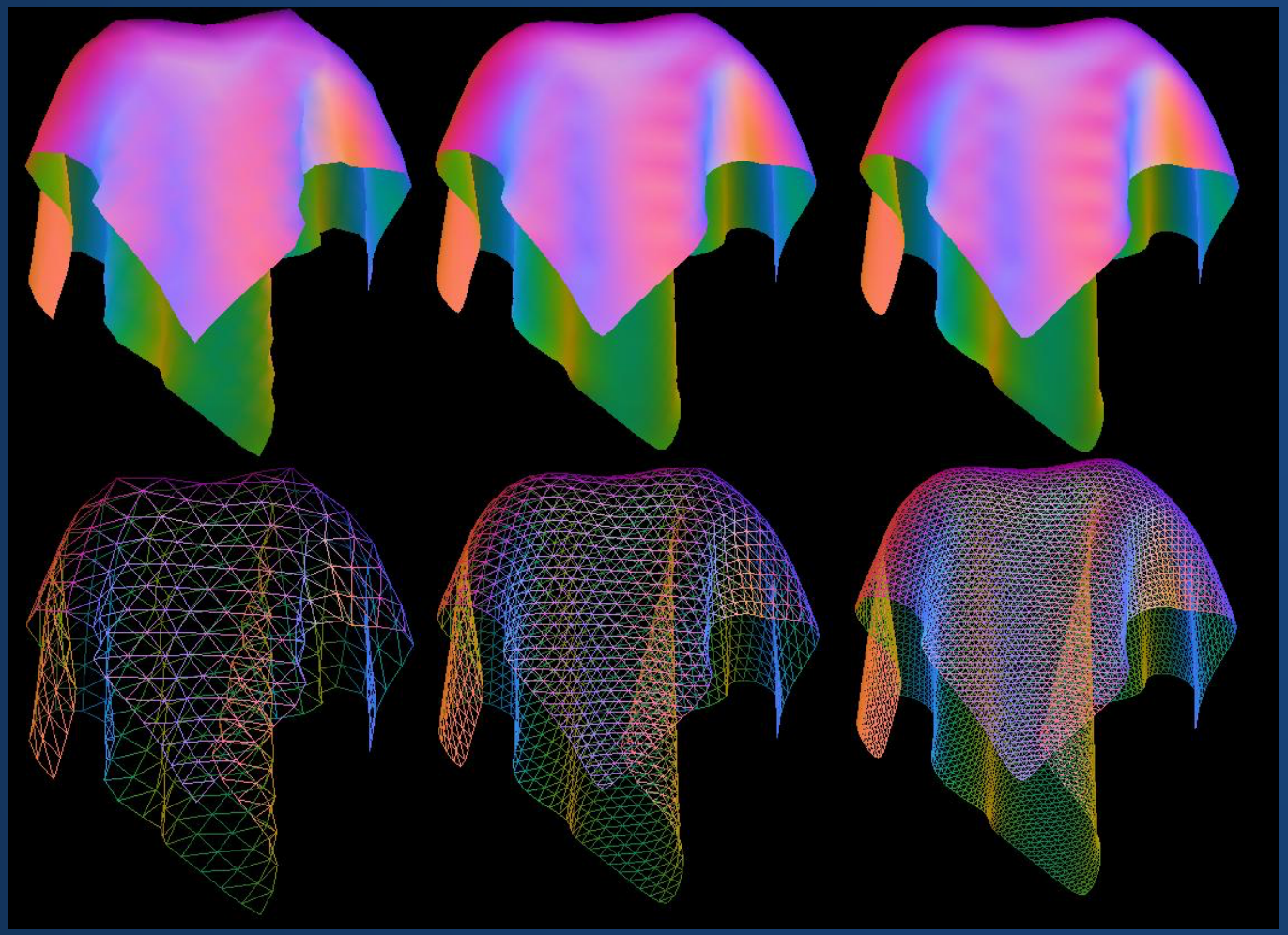


Are all (physical) simulations just really complex approximations of real equations?

Yep. Finding out which ones work and which ones don't is a complicated dance of mathematics, computer science, and just trying stuff out to see what works and what doesn't.

How do modern games create such realistic physics? How many points do you need to make realistic cloth?

Tons of work and engineering. Surprisingly, you don't need that many points to make a good cloth, but you need **lots** of springs.



If I wanted two things to stick together on collision, how would I do that?

General rule for physical simulation: think of some constraint/relation you want your system to satisfy, then use that to write your physics.

What would be a good rule to apply in this case?

Can you demonstrate an example where the force is applied non-axially?

When will we learn about collisions?

Announcements

Week 5 (Last Week)

Final Projects

Moving away from physical simulation

Previously, all our simulations were *physical* in nature, i.e. we were trying to capture some essence of physics.

However, we can also simulate things which don't directly involve physics.

These are "rules-based" simulations.

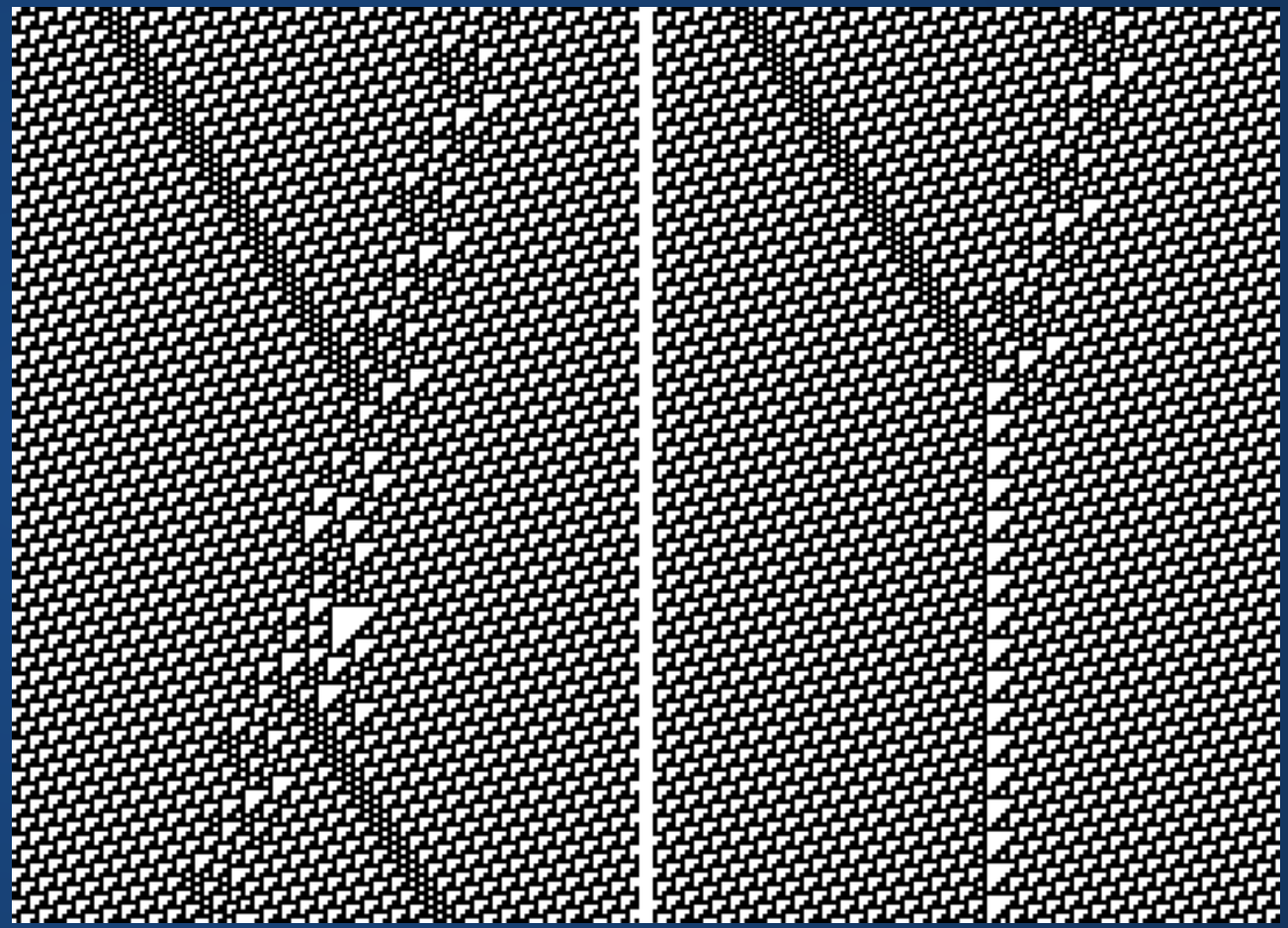
For example, many games can be simulated using rules-based simulations.

Stockfish (white) vs GPT-3.5 (black) in the most hilarious game of all time.

Cellular Automata

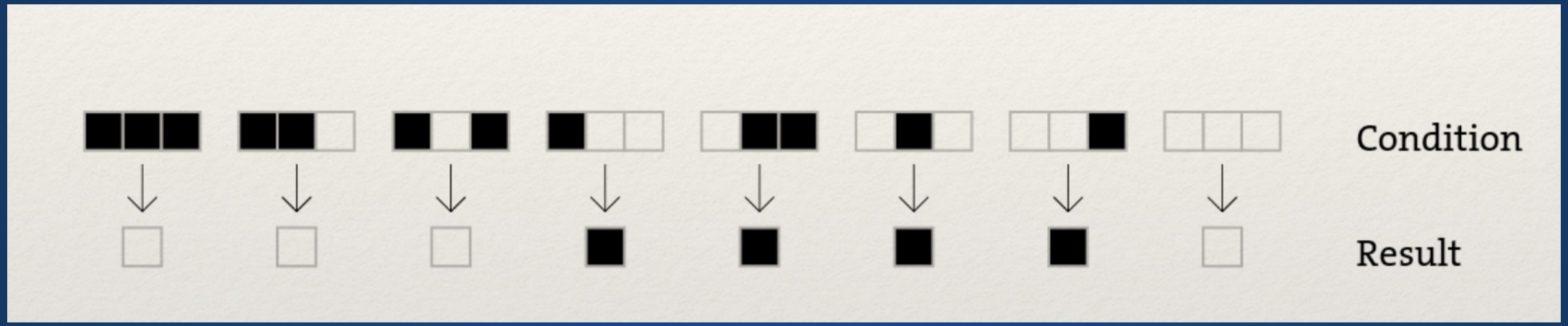
Cellular Automata

- A very popular type of simulation based on a grid.
- Rules tend to be of the form "if `<condition>` is true about neighboring cells, then do `<x>`.
- Observation: very simple rules can lead to complex patterns.

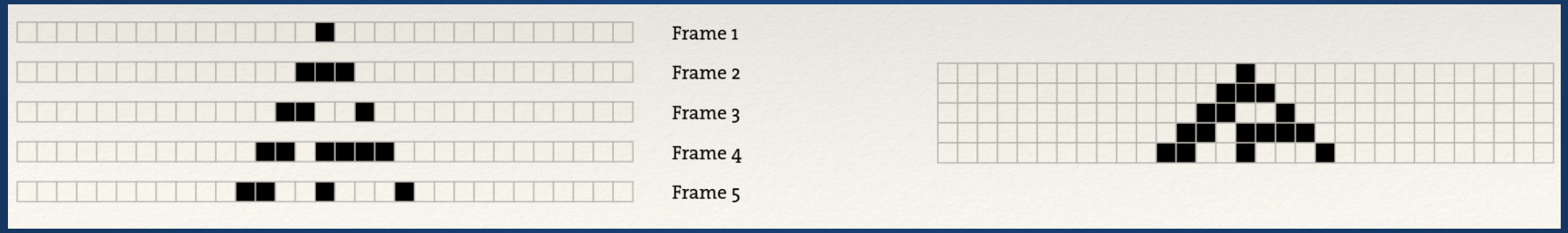


One-Dimensional CAs

Cells are in a row. Each cell has two neighbors. The cell's state in the next timestep depends on its current state and its neighbor's states.



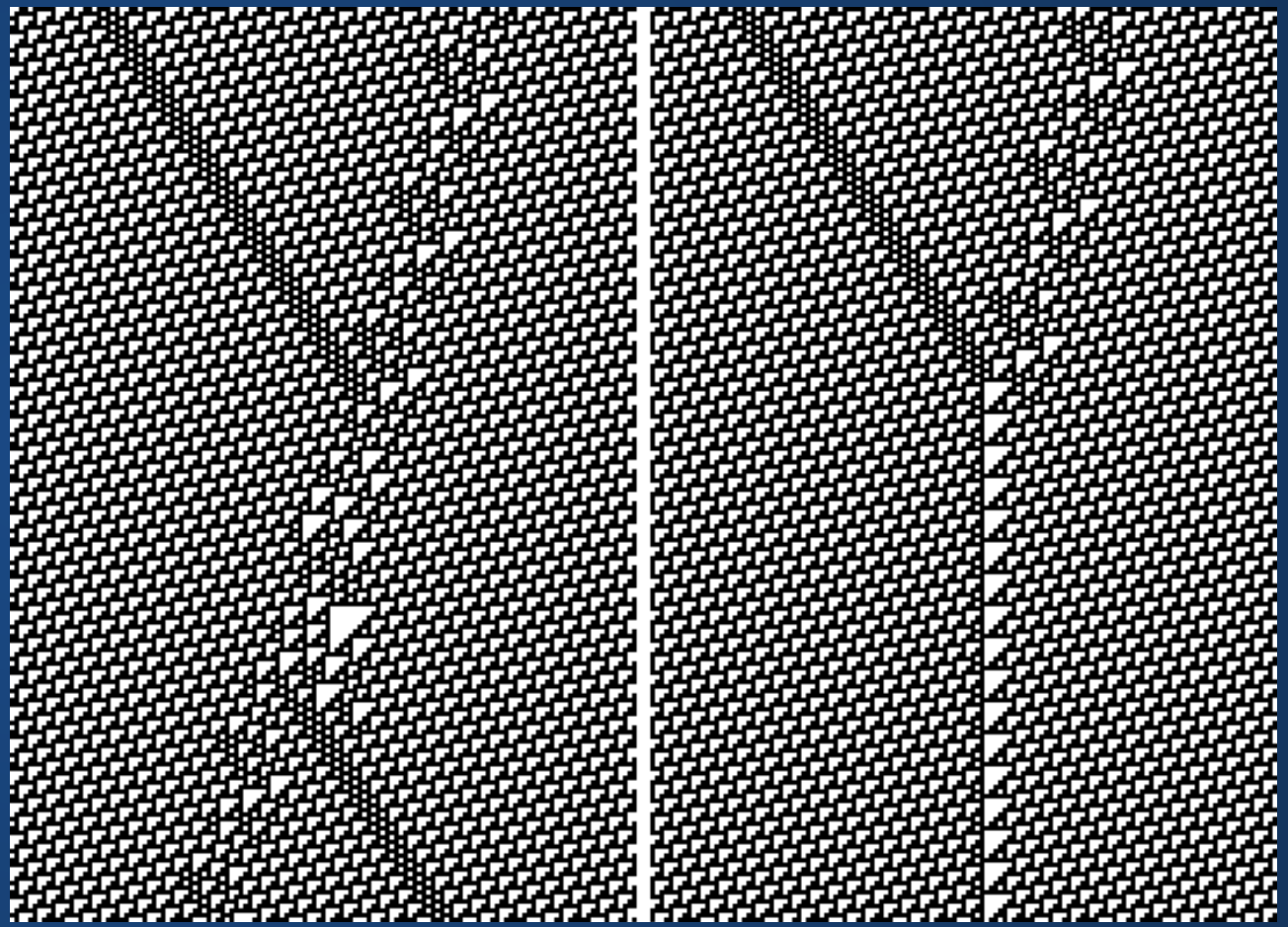
We can stack changes in various frames to form 2D images:



Fun Fact

Even one-dimensional cellular automata have *universal* computing power!

The automata shown is known as "Rule 110". It has been proven that anything that is computable on a "real" computer can also be computed using Rule 110.



Conway's Game of Life

Conway's Game of Life

- A two-dimensional CA developed by John Conway in 1970.
- Value of cells defined by values of neighboring cells
- Cells have two states: alive or dead

John Conway, 1937-2020

At each step:

- A cell with <2 neighbors will die, as if by loneliness
- A cell with 2-3 neighbors will survive if it is alive
- A cell with >3 neighbors will die, as if by overpopulation
- A dead cell with exactly 3 neighbors will be born, as if by reproduction

Simulating the Game of Life

- Each cell is represented by an integer (0/1) or boolean (true/false) value.
- State of the game is represented by an array of integers
- Apply update rules to each cell to complete a timestep.

Question: can we write the new state of a cell directly back to the array? Why or why not?

```
1  grid[i, j] = conway_rules(grid, i, j)
```

https://www.youtube.com/embed/xP5-iIeKXE8?enablejsapi=1

Autonomous Agents

Agents

Agent-based systems have autonomous agents which sense their environment and act on it. Agents have:

1. Environmental input
2. Rules for interacting with the environment
3. Goals

Even without global structure or understanding, complex behaviors can emerge!

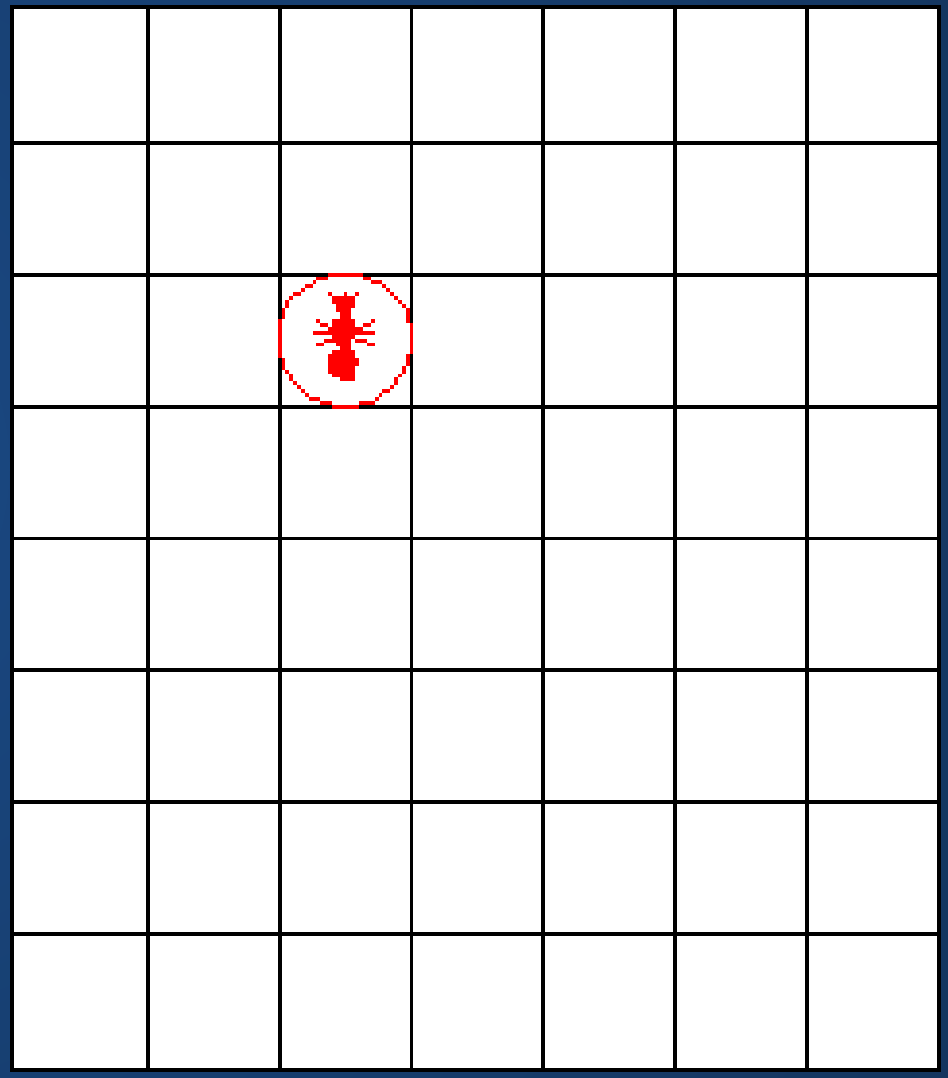
Langton's Ant

Agents are ants on a grid.

At every step, an ant follows these rules:

1. Move one grid cell forward
2. If pixel is white, make it black and turn right
3. If pixel is black, make it white and turn left

Also a universal computer (of course...)



From Wolfram MathWorld

Langton's Ant

Assume ant direction maps to one of the directions/ ints SOUTH (0), EAST (1), NORTH (2), and WEST (3).

What does this code do to the ant?

```
direction = (direction-1) % 4;
```

Gridless Autonomous Agents

Flocking

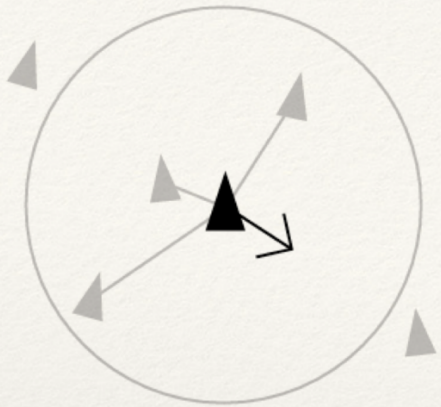
Oftentimes, small flying or swimming animals (birds, insects, fish) will form groups which seem to develop their own movement rules.

However, it is extremely unlikely that all the members are receiving telepathic commands from a leader.

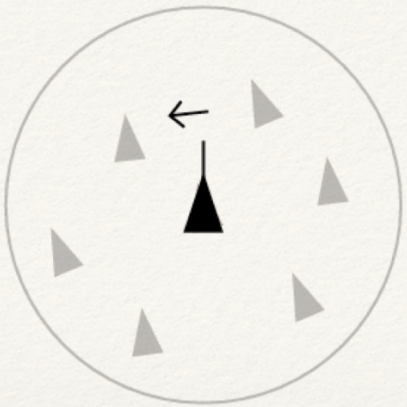
BOIDS

A gridless agent-based simulation of flocking which assumes that every member of the flock tries to meet three rules:

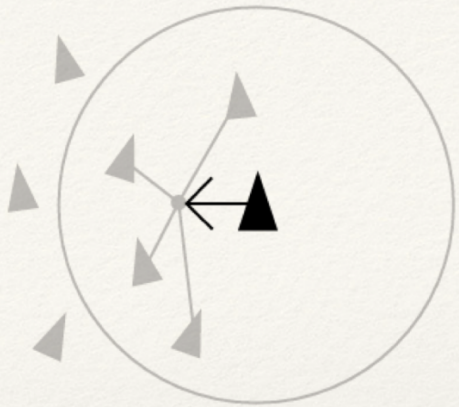
1. *Separation*: try to avoid collisions with local neighbors
2. *Alignment*: try to line up direction with the average direction of local neighbors
3. *Cohesion*: try to move to the average of the local neighbor's positions



Separation:
Steer to avoid crowding local flockmates



Alignment:
Steer toward the average heading of local flockmates



Cohesion:
Steer to move toward the average position of local flockmates

1. Get current position
2. Get position and direction of local flockmates
3. Calculate new heading and position based on local neighbor data
4. Update position + heading

Example: Predator-Prey

https://www.youtube.com/embed/rN8DzlgMt3M?enablejsapi=1

Note: video is from 15 years ago, so quality is a little low

Hands-On: Cellular Automata

Implement a simple version of one of the following:

1. Conway's Game of Life. Patterns may quickly appear and vanish, so consider either slowing the framerate or using interactivity to update the world (e.g. spacebar to draw the next frame)

2. Langton's Ants. In this case, it may take some time for a complex pattern to emerge.

drawGrid.pde on Canvas may help with
the rendering side of this.

Index Cards!

1. Your name and EID.

2. One thing that you learned from class today. You are allowed to say "nothing" if you didn't learn anything.

3. One question you have about something covered in class today. You *may not* respond "nothing".

4. (Optional) Any other comments/questions/thoughts about today's class.