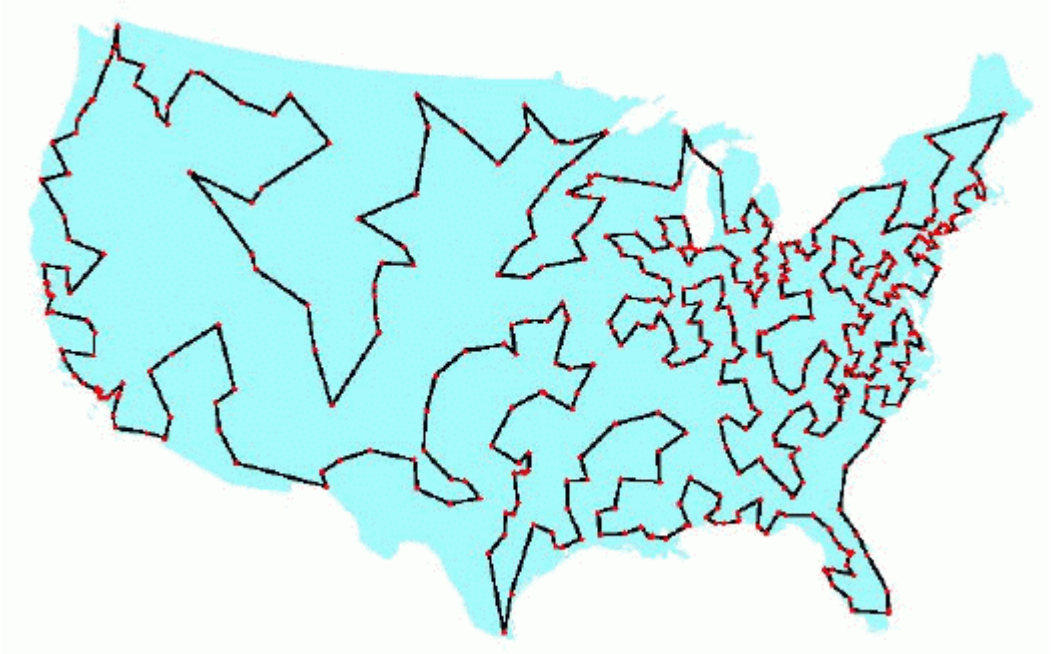
Traveling salesman

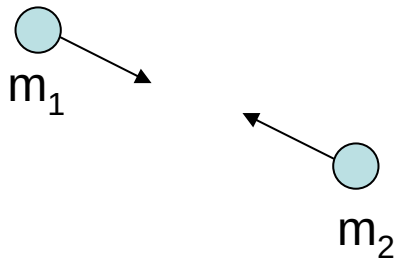
- Classical optimization problem
  - Visit *n* cities, return to the starting point
  - Minimize total distance traveled
- High complexity (n!), requires heuristics in order to be solvable in reasonable time



N-Body simulation

- Simulate the gravitational attraction of multiple bodies

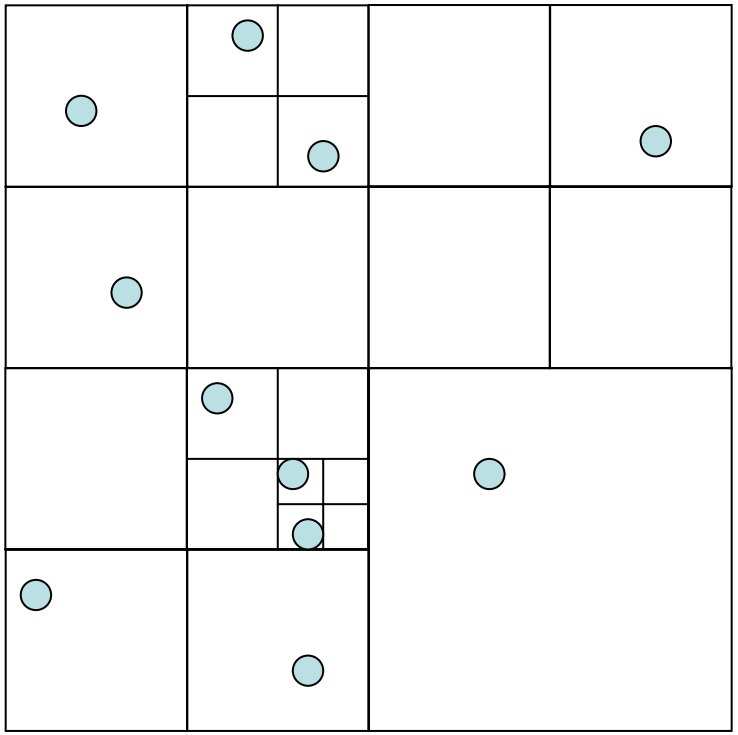
$$F = G \cdot \frac{m_1 \cdot m_2}{r^2}$$



- Calculate mutual influence for large numbers of bodies

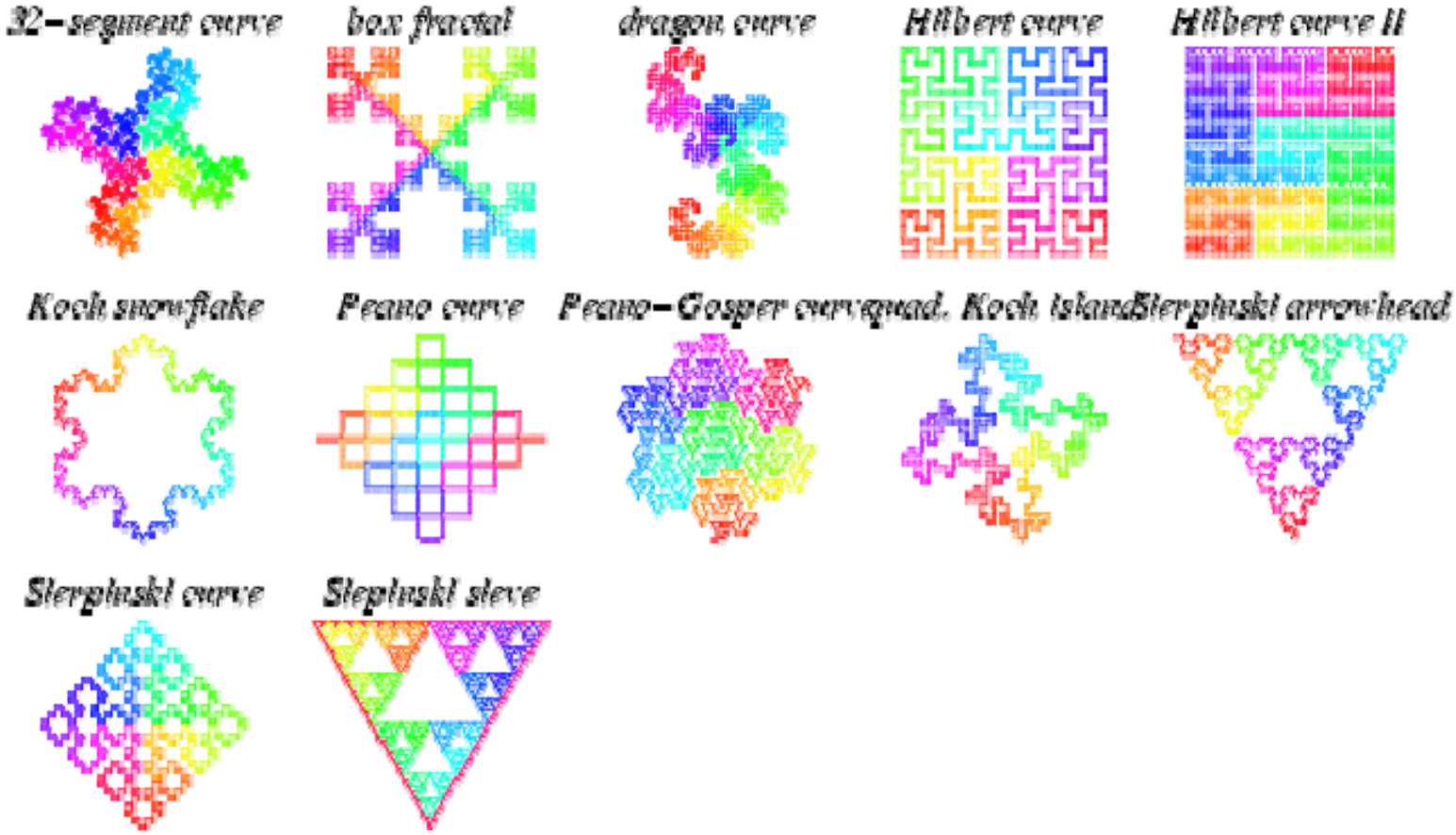
N-body optimization

- Replace individual distant bodies with clusters
  - The mass of a cluster is the sum of the masses of the bodies
  - The position is the center of gravity
- Use a quadtree for clustering



Lindenmayer Systems (L-Systems)

- String rewriting system used for generating fractals



L-Systems

- Simple grammar for describing movements:

  F: Forward +: Rotate left -: Rotate right

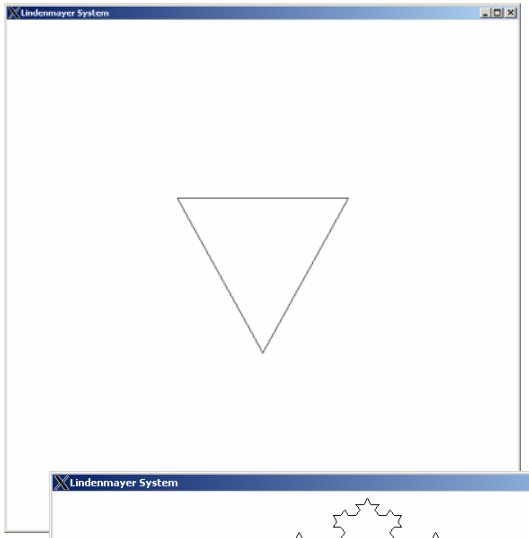
  [: Store current position and angle

  ]: Return to last stored position and angle

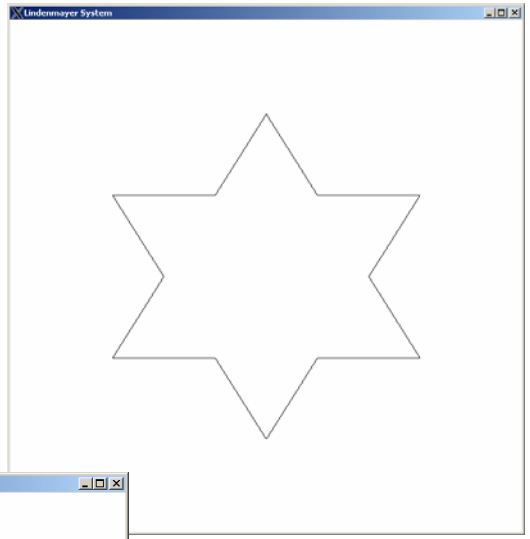
- Start with a base string, and use replacement rules
- Example: Koch Snowflake
  - Base: F--F--F, angle = 60°
  - Rules: F → F+F--F+F

Koch snowflake

Base (0): F--F--F

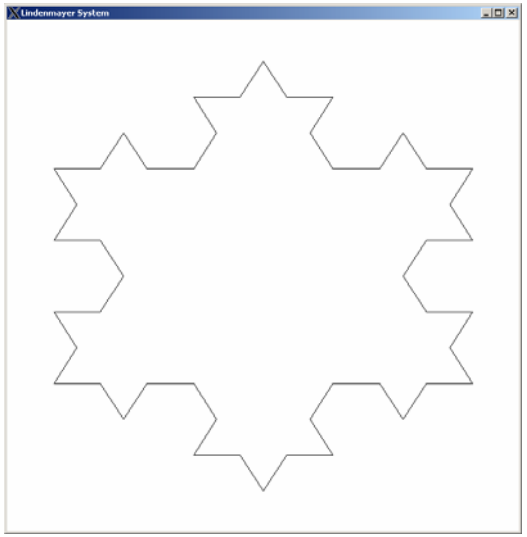


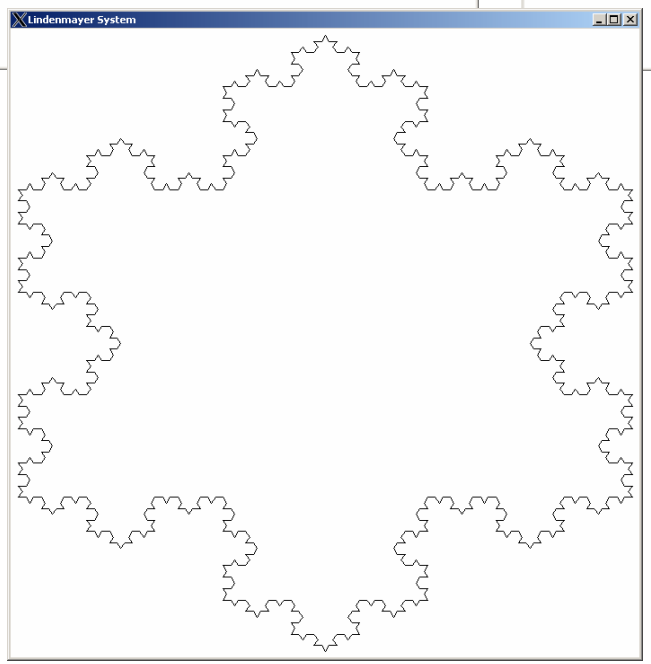
1: F+F--F+F--F+F--F+F--F+F--F+F



2:

F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F--
F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F--
F+F--F+F+F+F--F+F--F+F--F+F--F+F+F+F--F+F





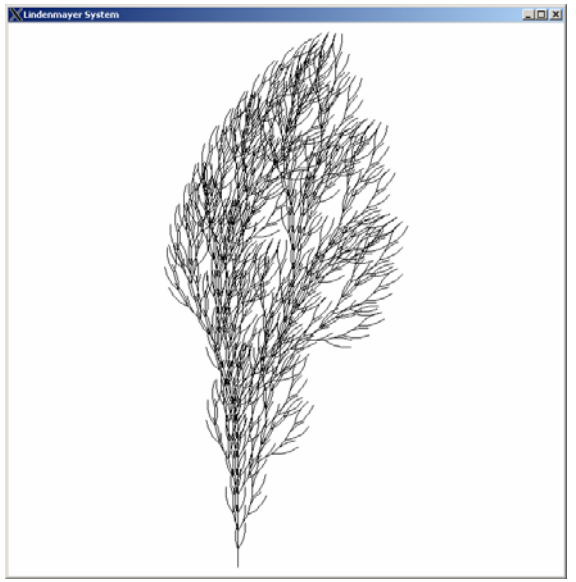
4:

F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F+F+F--F+F+F+F--F+F--F+F--F+F--F+F+F+F--F+F--F+F--
F+F+F+F--F+F--F+F--F+F+F+F--F+F+F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F+F+F--F+F+F+
F--F+F--F+F--F+F+F+F--F+F+F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+
F--F+F--F+F+F+F--F+F+F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F--F+
F--F+F+F+F--F+F+F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F--F+
F+F+F--F+F+F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F+F+F--F+F+F+F--F+F--F+F--F+F+F+F-
-F+F+F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F+
F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F+F+F--
F+F+F+F--F+F--F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F+F
+F--F+F--F+F--F+F+F+F--F+F+F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F+F+F--F+F+F+F--F+
F--F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F+F+F--F+F+F+F--F+F--F+
F--F+F--F+F+F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F--F+
F+F+F--F+F+F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F+F+F-
-F+F+F+F--F+F--F+F--F+F+F+F--F+F+F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F+F+F--F+F--
F+F--F+F--F+F--F+F+F+F--F+F+F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F+F+F--F+F--F+F+F
+F--F+F--F+F+F+F--F+F+F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F+F+F--F+F--F+F--F+F+F
+F--F+F+F+F--F+F+F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F--F+F--F
+F+F+F--F+F+F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F--F+F+
F--F+F+F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F+F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F
+F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F+F+F-
-F+F+F+F--F+F--F+F--F+F+F+F--F+F

Other L-systems

- Use position stack
  - Base: F, angle = 15°
  - Rules:

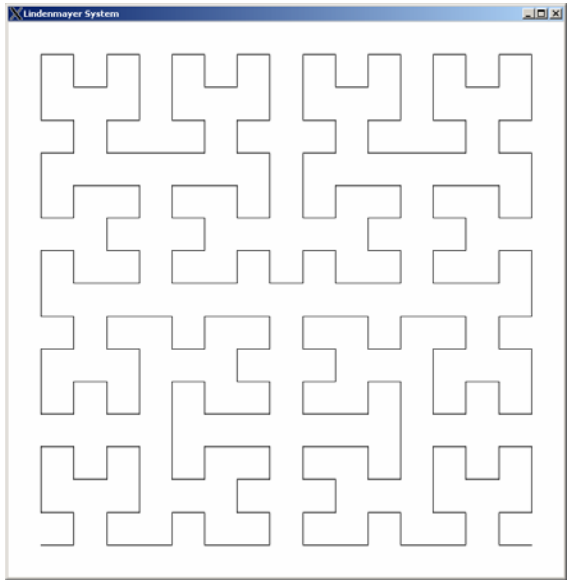
  F → FF[--F+F+F][+F-F-F]



- Use alternate characters
  - Base: L, angle = 90°
  - Rules:

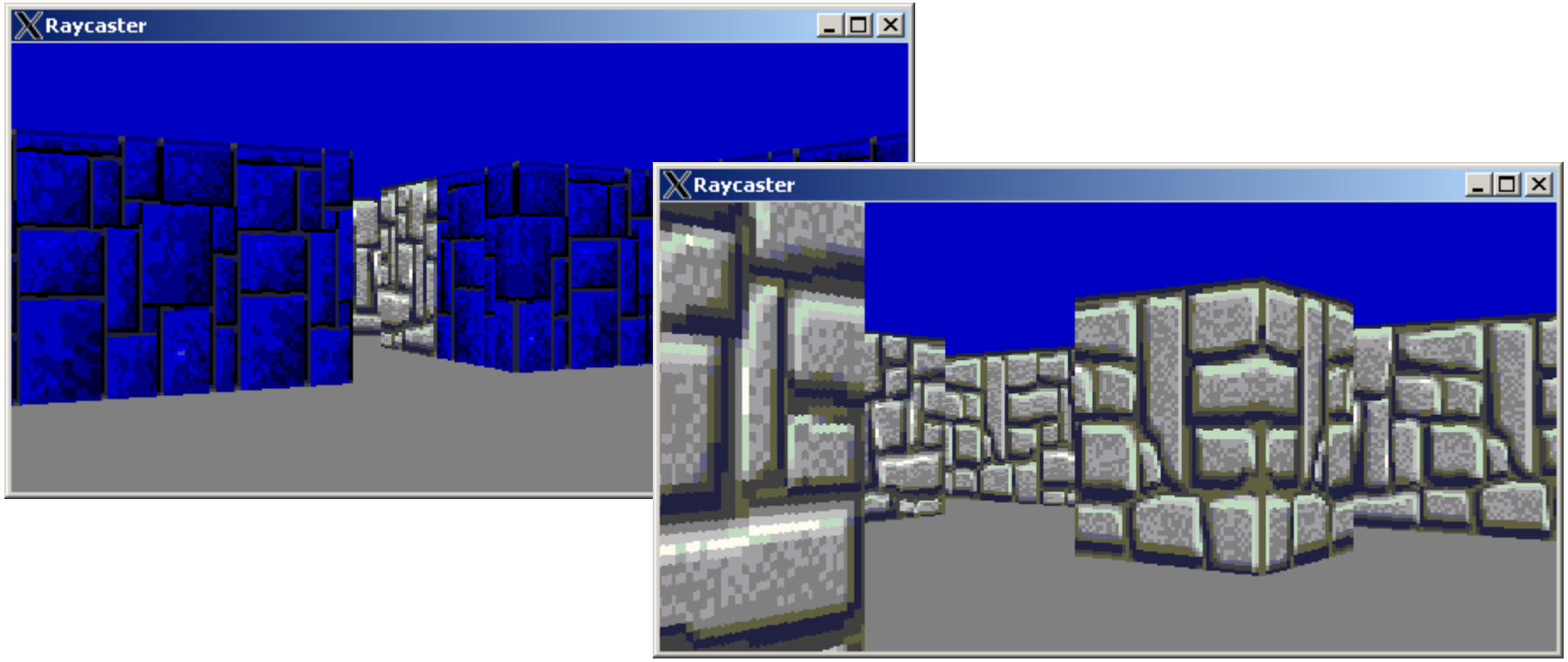
  L → +RF-LFL-FR+

  R →-LF+RFR+FL-

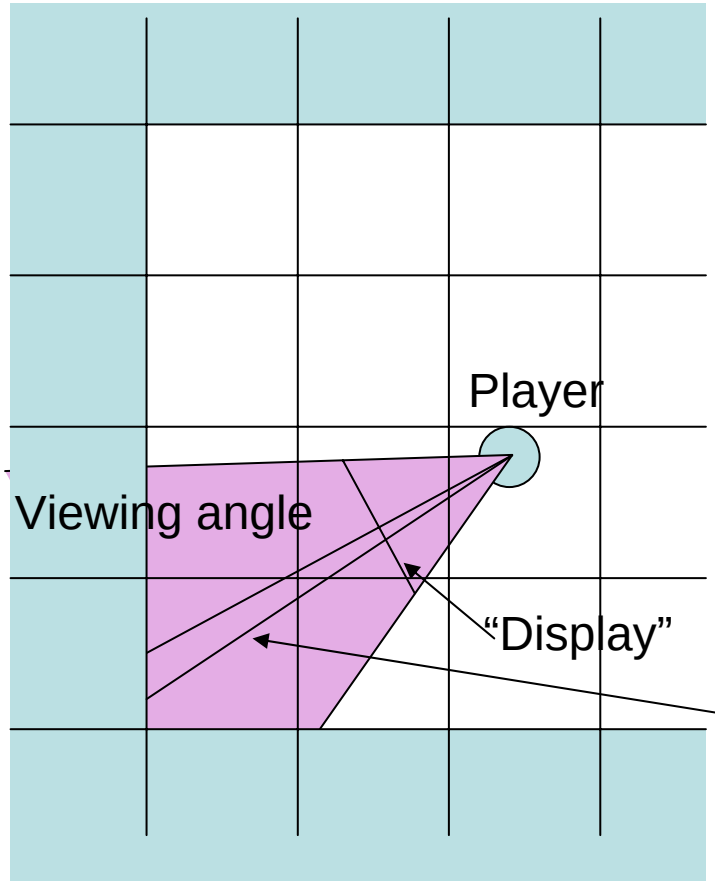


Raytracing in 2(½) Dimensions

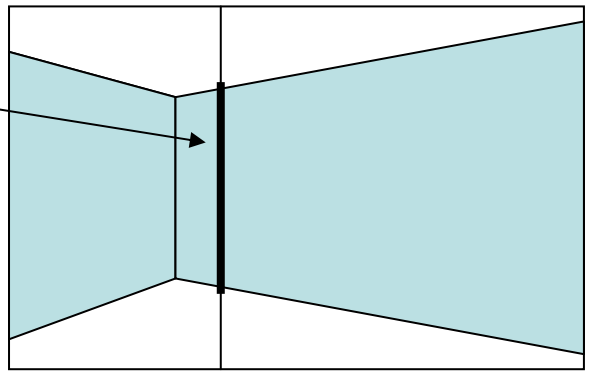
- Oldschool game engine technology
  - Used in Wolfenstein 3D (1992)
  - Later refined in Doom (1994) and Duke Nukem 3D (1996)
- Pseudo-3D environments



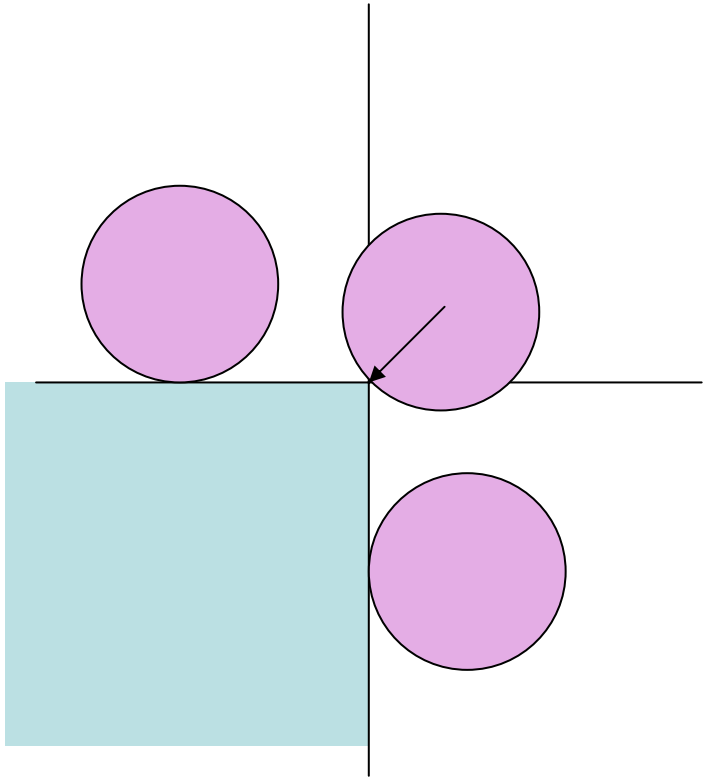
Raytracing



- For each column of pixels on the display, cast one ray into the map
- Find intersection point with walls
- Draw a column of height proportional to 1/distance
  - Apply scaled column of texture if desired



Basic gameplay mechanics



- Provide simple interactive movement
- Implement collision detection
  - Walls (circle/line)
  - Corners (circle/point)