Introduction

- Lonpos Pyramid is an engaging puzzle toy originating from Taiwan.
- It consists of 12 colorful blocks that can be assembled into either 2D or 3D pyramids.
- Each block is made up of 3 to 5 unique spherical components.

12 colorful blocks

2D pyramids

3D pyramids

Introduction

- **How to play?**
  - The toy designer provides puzzles specifying certain block positions that cannot be moved. Players must use the remaining blocks to construct the pyramid.
  - In the 3D pyramid, blocks may be positioned upright, making it more challenging than the 2D pyramid.

Algorithm Overview

1

Step 1

Identify all possible placements for each block in the pyramid and store them in a lookup table.

2

Step 2

Use backtracking algorithms to find solutions based on preset block positions.

Determining Block Positions in Space (1/3)

Step 1: Define the spatial positions of each sphere within the blocks.

Methods:

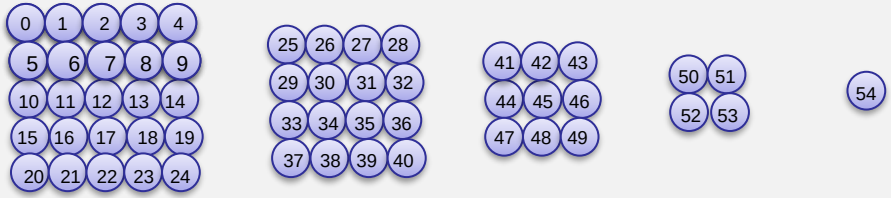
① Assign numerical identifiers (0-54) to each sphere.

② Map these identifiers to spatial coordinates (x, y, z).

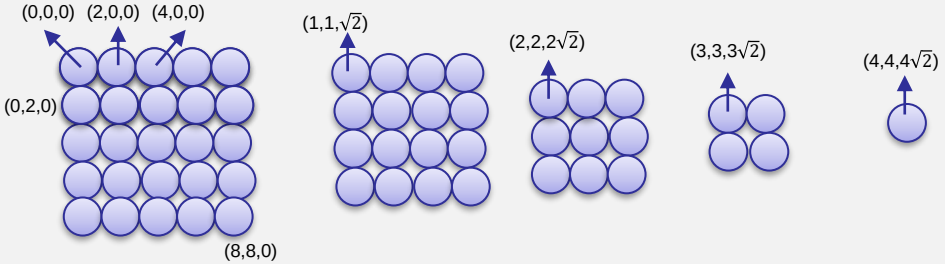
- Set the grid size to 2 for the x and y directions, and $\sqrt{2}$ for the z direction.

③ Input the identifiers of any two spheres to obtain their distance.

Identifiers: 0~54



Coordinate (x, y, z)



Determining Block Positions in Space (2/3)

Step 2: For blocks with **fewer than 5** spheres, use a **brute-force method** to identify all possible placements.

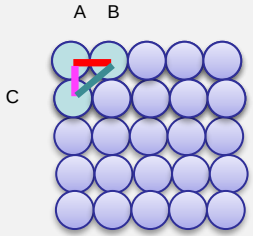
Methods:

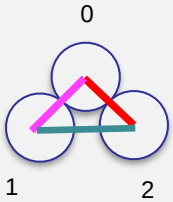
① Select any three spheres from a pyramid of 55 spheres.
② Calculate the distances between each pair of the selected three spheres.
③ Check if these distances match the distances between any two spheres in a white block.
④ If they match, it indicates that the selected three spheres form a shape identical to that of the white block.

Diagram

Take 

as an example:





- Distance (A , B) = Distance(0 , 2)
- Distance (A , C) = Distance(0 , 1)
- Distance (B , C) = Distance(1 , 2)

Determining Block Positions in Space (3/3)

Step 3: For blocks with **exactly 5 spheres**, combine placements of **smaller blocks. (divide and conquer)**.

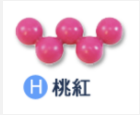
Methods:

① Consider the pink block as a combination of 2 white blocks.

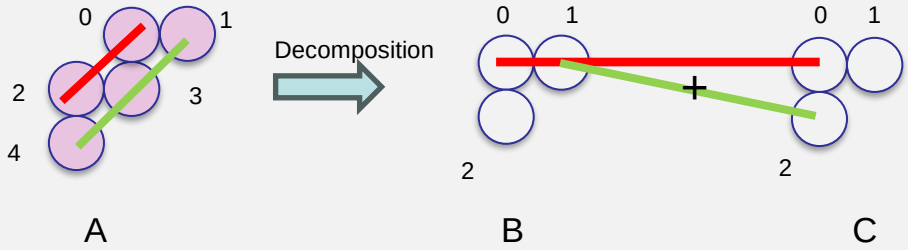
② Reference the lookup table and calculate all combinations of the 2 white blocks.

③ Each combination must be checked to meet the restrictions.

Restrictions

Take 

as an example:



- B2, C1 must be in the same position
- Distance (A0 , A2) = Distance(B0 , C0)
- Distance (A1 , A4) = Distance(B1 , C2)

Solution Finding

Step 1

Read user-inputted puzzle settings and filter out impossible block placements.

Step 2

Prioritize placing blocks with fewer placement options into the pyramid recursively.

Step 3

Use the backtracking algorithm to recursively find solutions.

Each recursion must first check if the current placement of the block can be placed into the pyramid.

- **If it can be placed**: Insert it into the pyramid, then sort the remaining blocks that have not yet been placed by the number of possible placements, and continue recursively with the block that has the fewest placements (**greedy approach**).
- **If it cannot be placed**: Try the next placement for the current block.
- **If the last block is successfully placed**: Return the result.

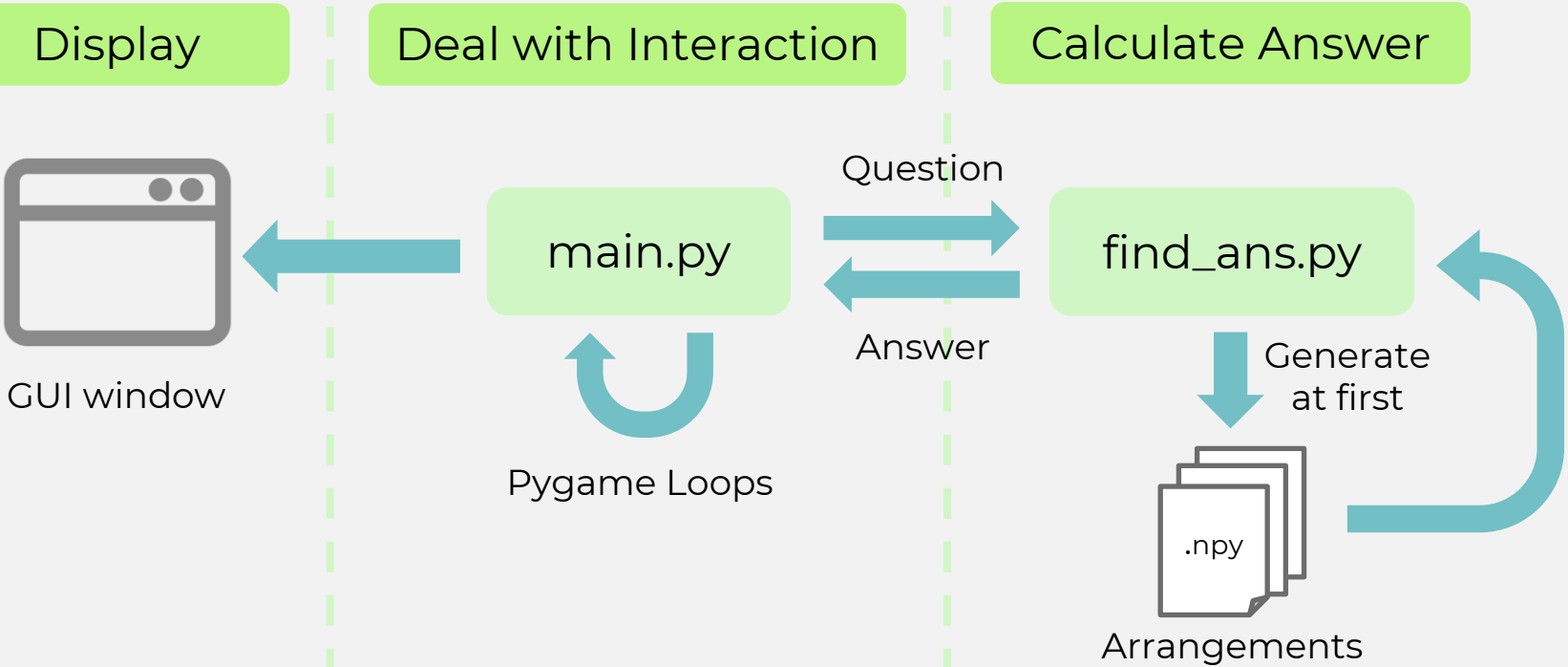
Conditions for a block to be placed into the pyramid:

① The intended position for insertion has not been previously occupied by another block.

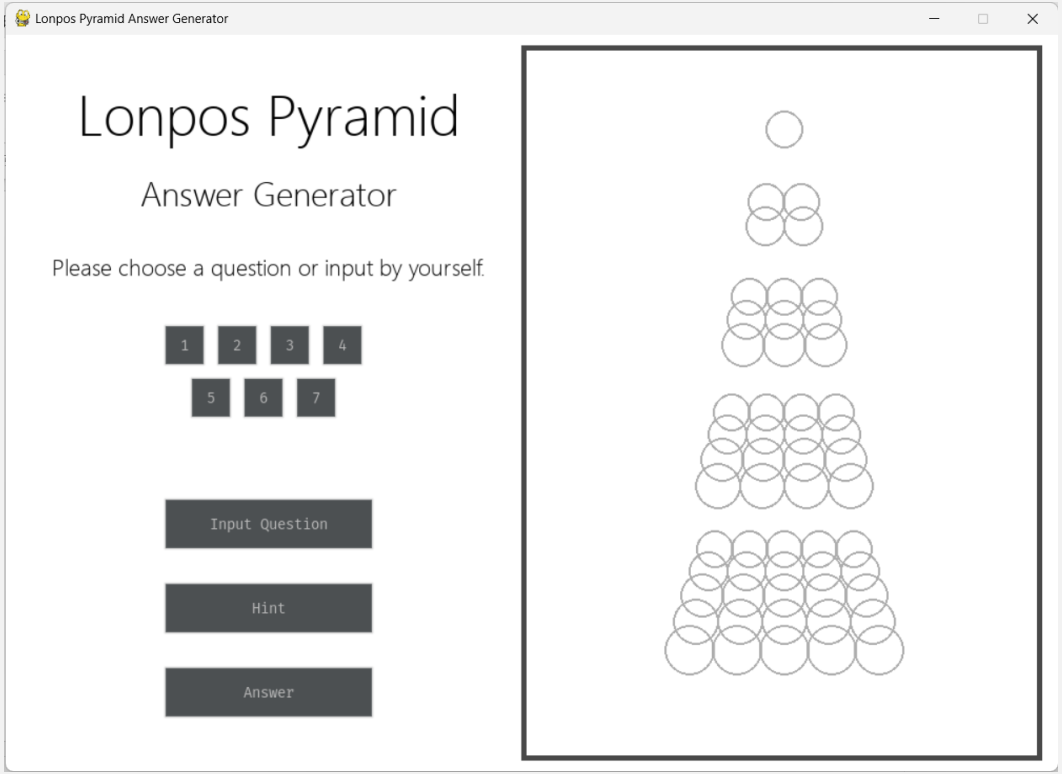
② After placement, check all possible placements for the remaining blocks not yet in the pyramid; ensure there is no sphere that no block can cover. If such a situation occurs, it indicates a dead end, and the path should be abandoned.

③ If all the above conditions are satisfied, sort all the remaining blocks that have not yet been placed and return them.

System Structure

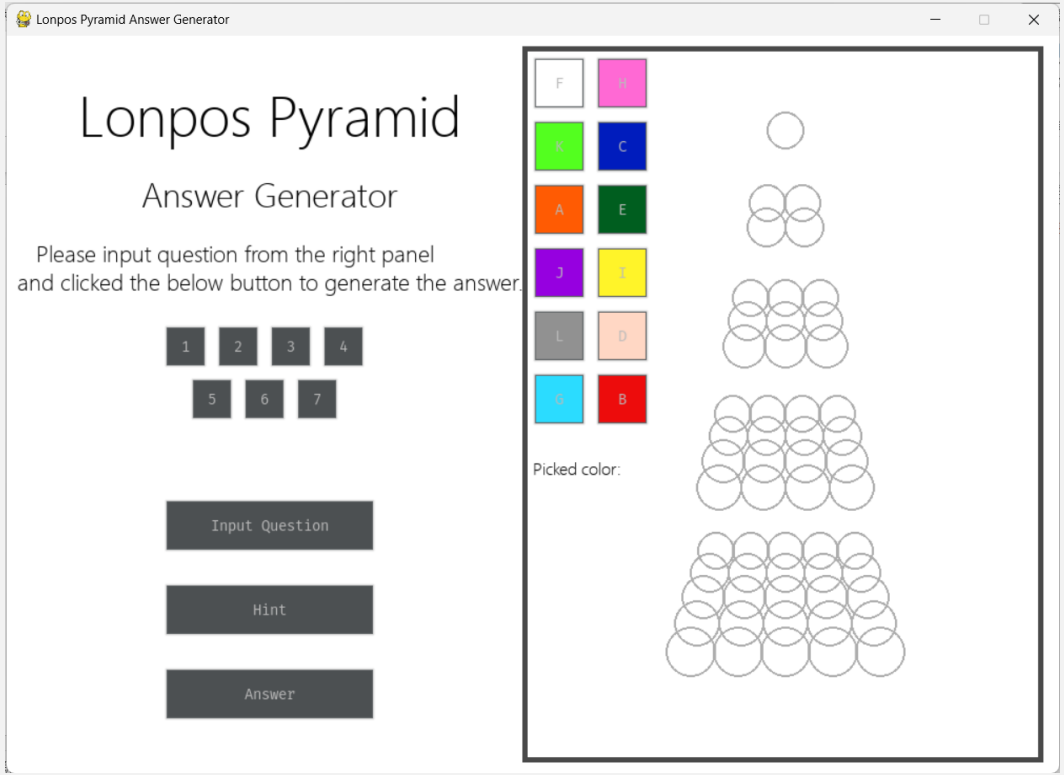


User Interface



- **1 to 7 Buttons**
  Press to choose question.
- **Input Question Button**
  Input question by yourself.
- **Hint Button**
  Generate the steps of building pyramid.
- **Answer Button**
  Directly show the final answer.

User Interface



Color picker

In the input mode, you can pick color you want by clicking the buttons.

Next, click the position to draw the picked color on the sphere.

After finish drawing, click the left side button and it will start calculating the answer.