Level 4 Special Project

Sponges

Prerequisite Concepts	Concept 12
Key Concepts	Concept 19

We will use the term *sponge* to refer to a porous three dimensional LEGO® artifact. One of the most famous sponges is the Menger sponge, shown in Figure 1.

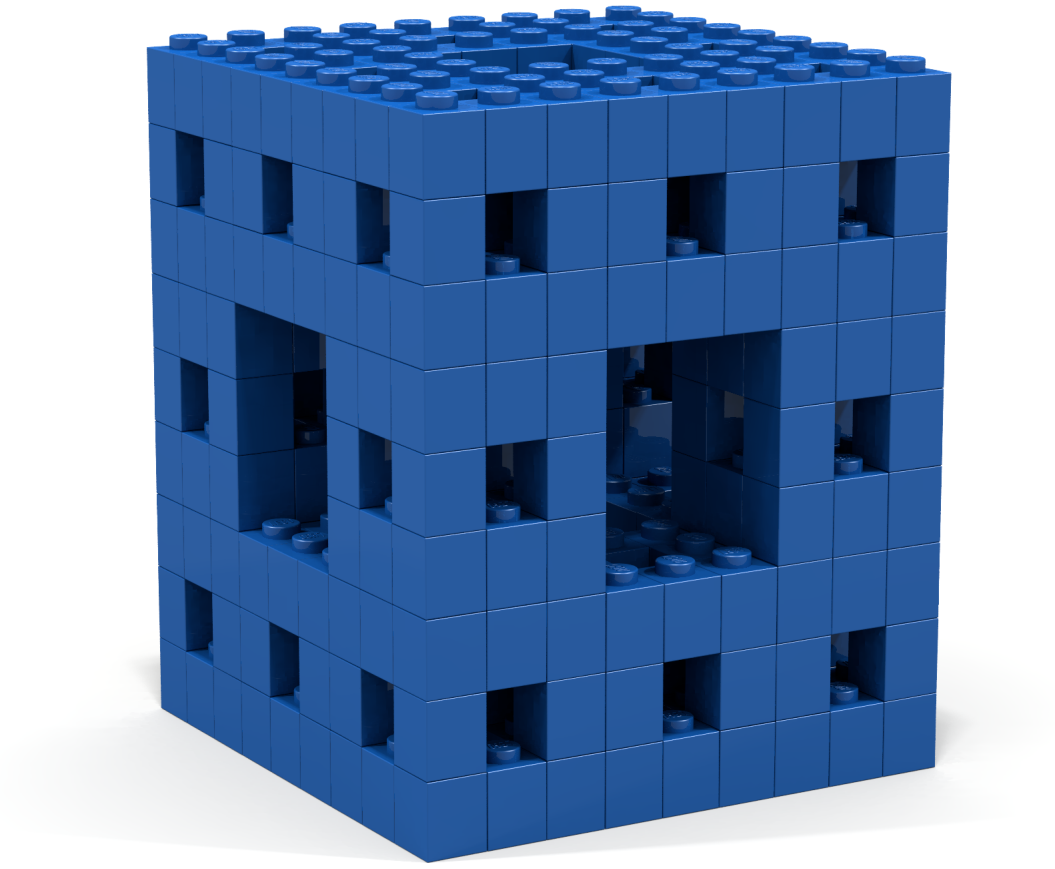
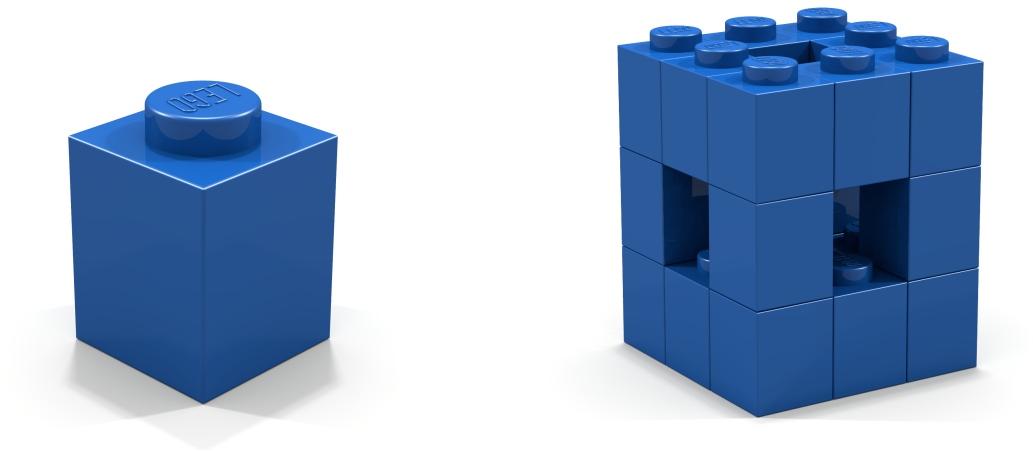


Figure 1: Level 2 Menger Sponge.

In general, a sponge can be created though the geometric repetition of a 3-dimensional pattern. Typically, the *seed* of a sponge is a bit-brick (i.e., a $1 \times 1 \times 1$ brick) shown in Figure **??**. Using this seed, an $n \times n \times n$ cube (or even a rectangular prism) is created with some pieces removed. The resulting structure is a sponge. Note that the creation of a $3 \times 3 \times 3$ sponge involves pattern spanning 27 bit-brick positions.

The Pattern for the Menger Sponge

Using a bit-brick as the seed a $3 \times 3 \times 3$ Menger sponge pattern is created by omitting 7 bit-bricks from a solid $3 \times 3 \times 3$ cube. Specifically, the center brick of each of the six faces of the cube is omitted as is the center of the cube itself. This pattern can then be geometrically repeated to create larger and larger sponges.



(a) Level 0 Menger Sponge

(b) Level 1 Menger Sponge

Figure 2: Constructing the Menger Sponge.

Write a program that creates a level 2 Menger sponge like the one shown in Figure 1.

Adding Color

Bricklayer provides the ability to define a function that, when called, will return a brick randomly selected from a list of bricks. The Pieces structure of Bricklayer defines a number of lists that can be used for random generation. Some of these lists are shown in Table 1.

List	Comment
grayScale	black, white, and all gray bit-bricks
greenScale	all green bit-bricks
blueScale	all blue bit-bricks
redScale	all red and pink bit-bricks
clearScale	all clear bit-bricks

Table 1: Some of the predefined brick lists from the Pieces structure.

The val-declaration below gives an example of how you could declare a function called *randomBrickFn*.

```
val randomBrickFn = generateRandomBrickFn blueScale
```

This function can be called as follows:

randomBrickFn()

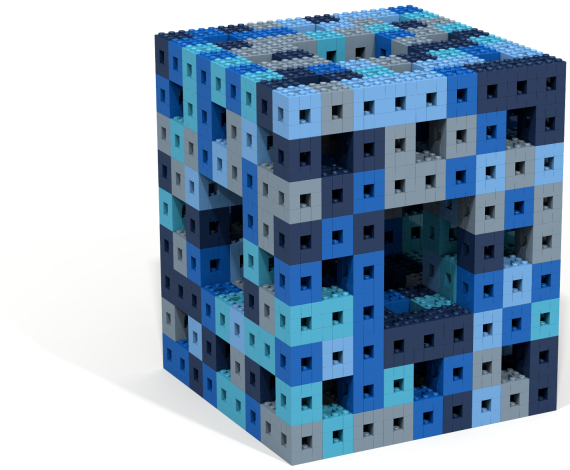
In this example, when called the function randomBrickFn will randomly select a brick from the list blueScale.

In addition to the lists defined in the Pieces structure, you can also define your own lists as shown in the example below.

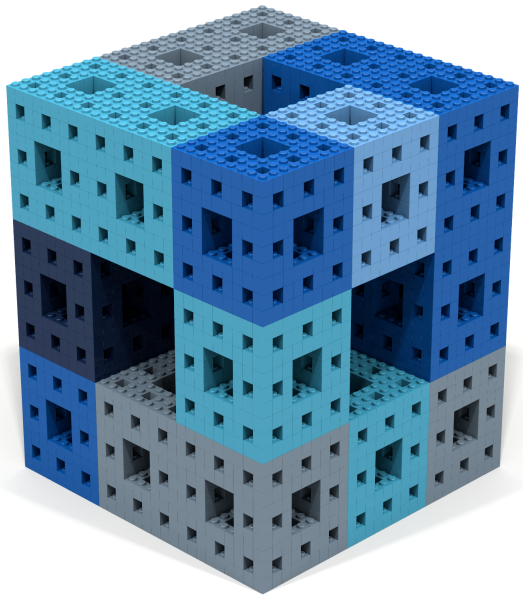
```
val myList        = [BLUE,RED,WHITE]
val randomBrickFn = generateRandomBrickFn myList
```

Sponges having interesting colors can be created by from generators that use random bricks to construct all sponges of a certain level. For example, a level 1 sponge could be generated using a single brick, but whenever a different level 1 sponge is to be generated the brick used is randomly selected. Figure 3 shows how such a uniform use of randomly generated bricks can be used to add color to the Menger sponge.

Create a Bricklayer program that builds a level 2 (or level 3) Menger sponge similar to the ones shown in Figure 3 .



(a) Level 1 brick uniformity



(b) Level 2 brick uniformity

Figure 3: Using random blueScale bricks to color the level 3 Menger sponge.