

# ASSIGNMENT 2

# ASSIGNMENT 2 GOALS

1. Saving the game state - storing everything needed to recreate the level
2. Traveling between levels - stairs
3. Procedural Modelling - creating an outdoor terrain
4. Animation - clouds

# SAVING THE GAME STATE

Everything that is part of the maze created in assignment 1 needs to be stored so when the player leaves the current level so they can return to it and find it is in the same state that it was when they left.

This includes everything in the `world[][][]` array that was created in assignment 1.

You can store all of the maze or you can recreate it.

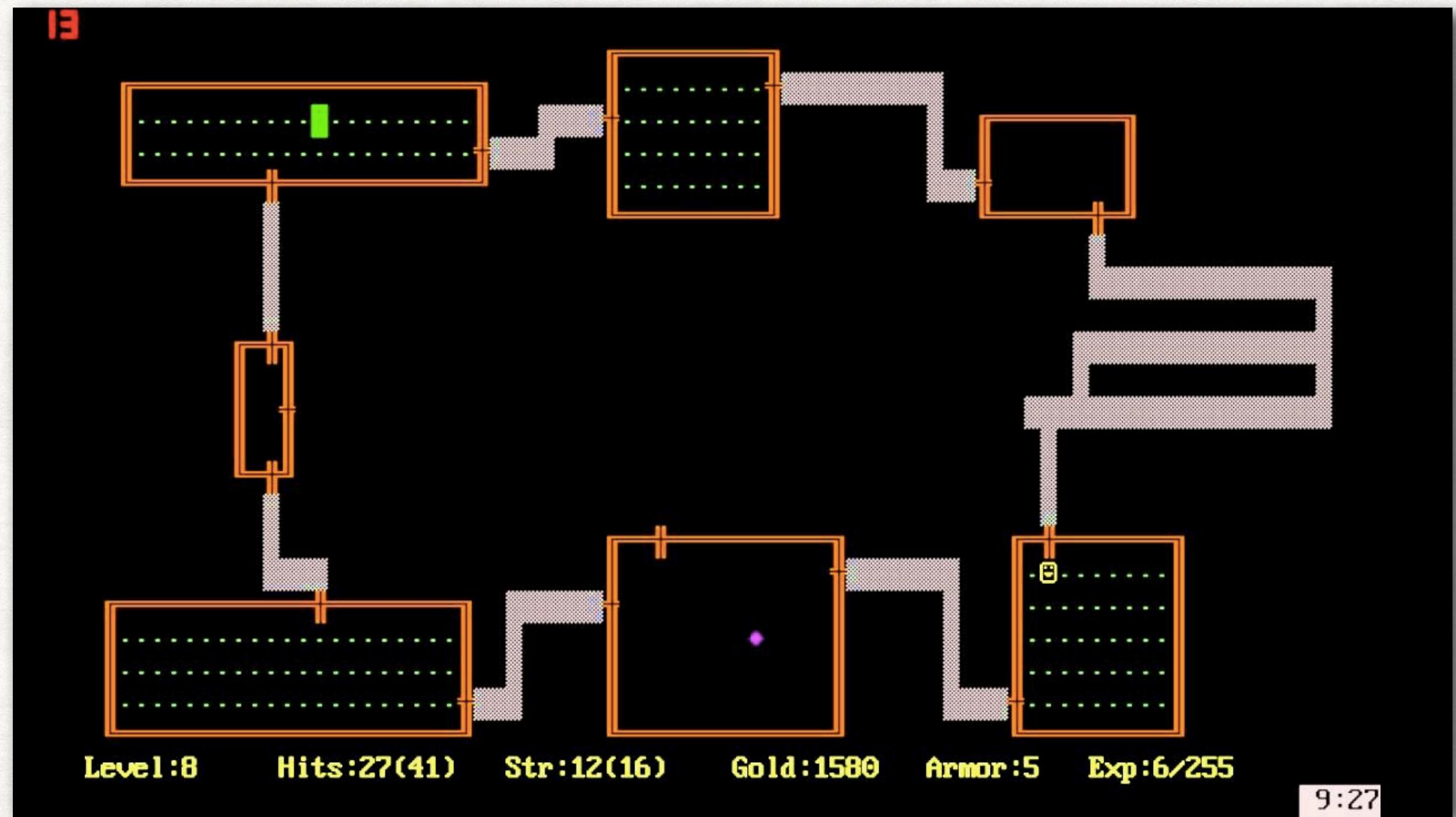


# SAVING THE GAME STATE

This currently includes the rooms, hallways, location of blocks on the ground, any texturing that places different coloured blocks in the walls, ceiling, or floor.

Assignment 2 requires stairs and an outdoor level to be added. These need to stored for each level.

More items will be added to later assignments that will be stored in each level. This will likely include treasure, monsters (which can move on their own), and possibly other items. Make your storage method flexible so other data types can be added.



# SAVING THE GAME STATE

You can choose to store the state in a data structure or a file.



# TRAVELING BETWEEN LEVELS

Rogue has multiple levels of maze and the player can go up or down between levels using staircases.

Stairs look like grey (down) and white (up) cubes. When a player stands on the cube they should automatically move up or down a level.

When a player leaves a level then all information from that level must be saved. The level they move to must either be created (if it has not been visited before) or must be loaded from the a saved state (if they have visited it before).



# TRAVELING BETWEEN LEVELS

Assignment 2 requires only two levels. Later assignments will require many more levels.

Place the staircase randomly in a room in underground levels.

Assignment 1 will require only one up staircase which leads to the outside level and one down staircase that leads to the underground level.

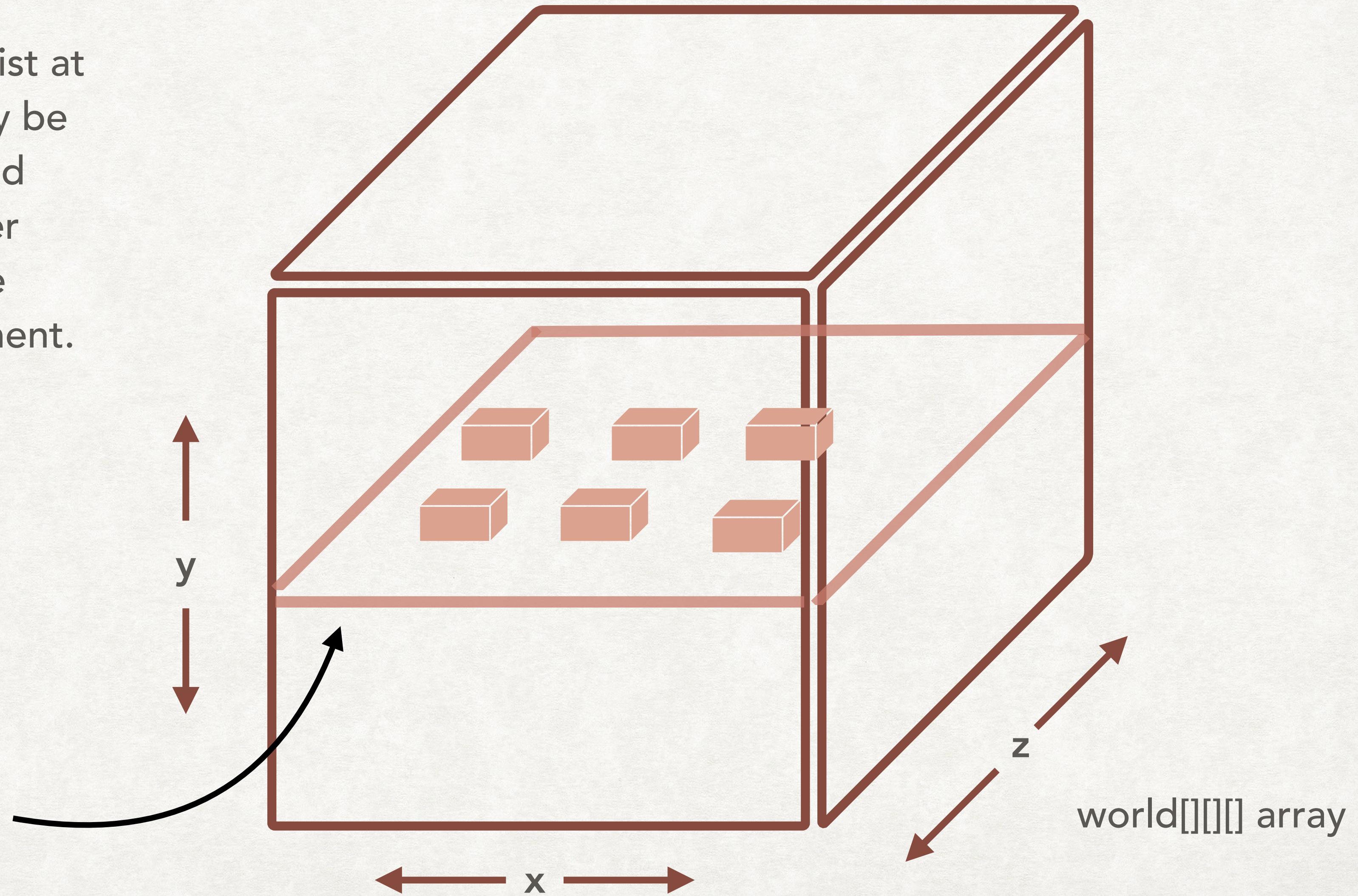
Later assignments will require more up and down staircases in the underground levels.



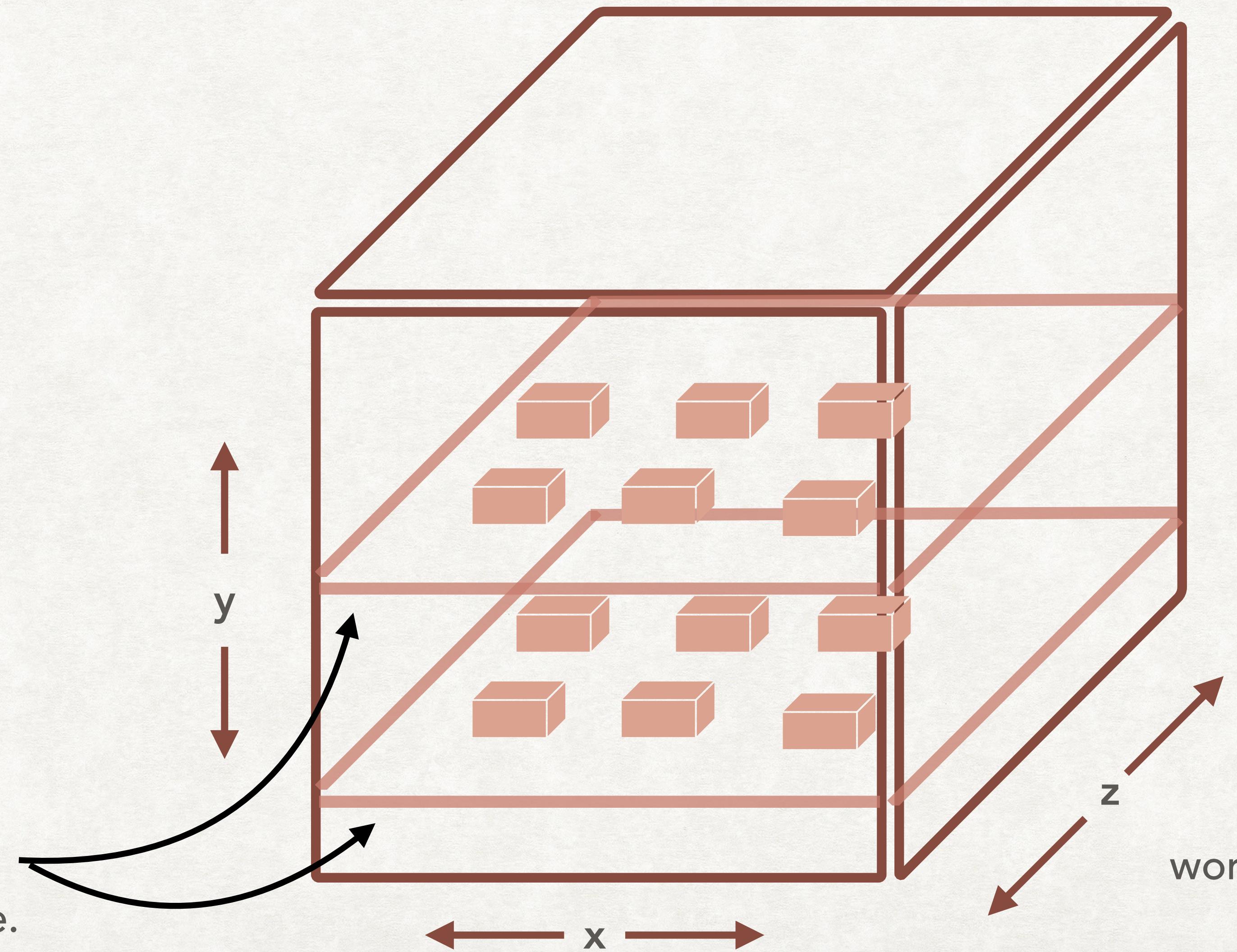
# TRAVELING BETWEEN LEVELS

Do not build all levels so they exist at the same time. There should only be one level in existence in the world array at a time. The state of other levels should be stored using the state system part of this assignment.

Only one level exists at a time in the world array.



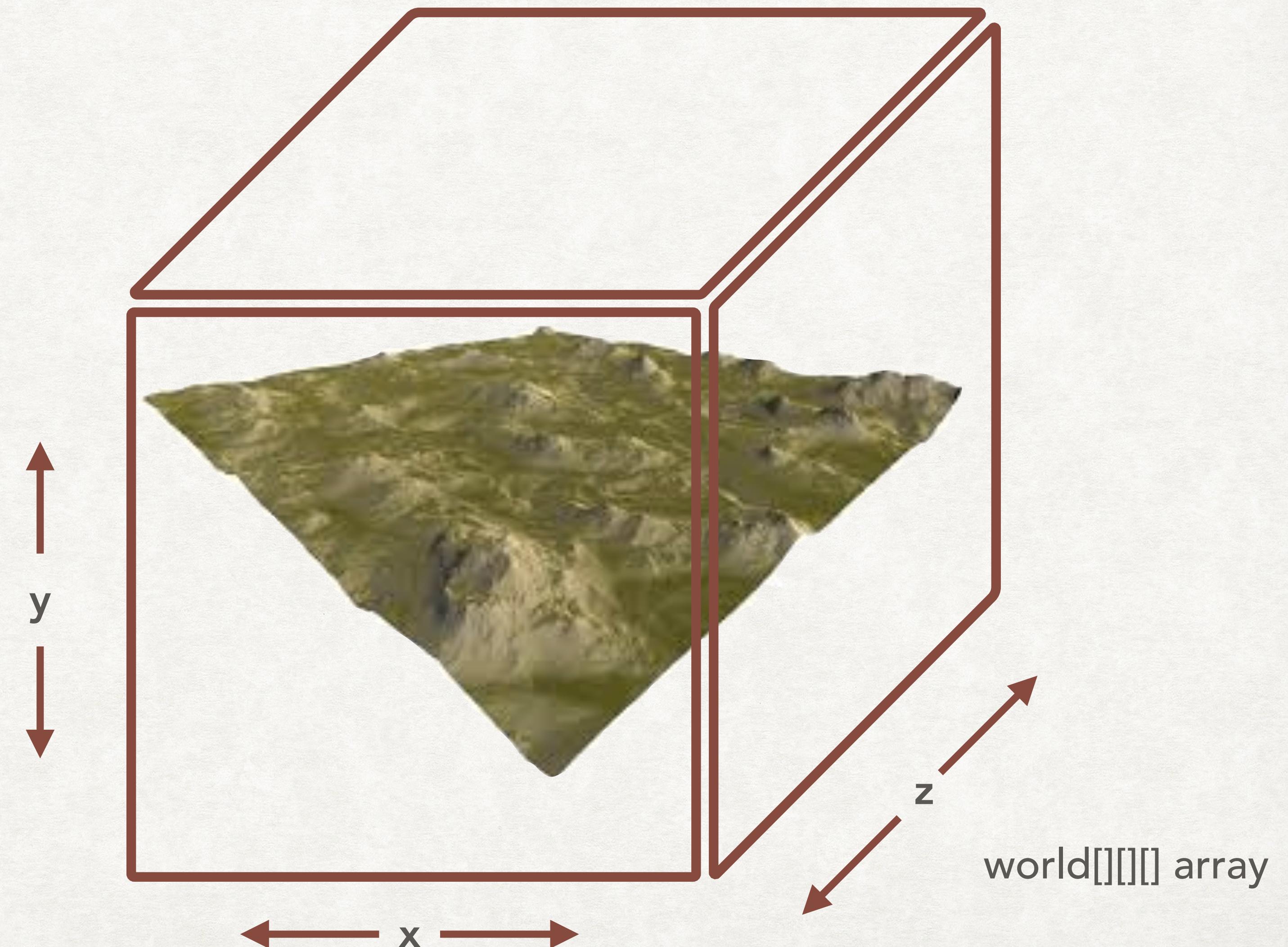
# TRAVELING BETWEEN LEVELS



More than one level at a time in  
the world array is not acceptable.

# PROCEDURAL MODELLING

Use Perlin noise to create an outdoor level for the game. The outdoor level is where the game will begin. Place the player near the down staircase in the outdoor level. The stairs should lead to a maze level when the player uses them.



# PROCEDURAL MODELLING

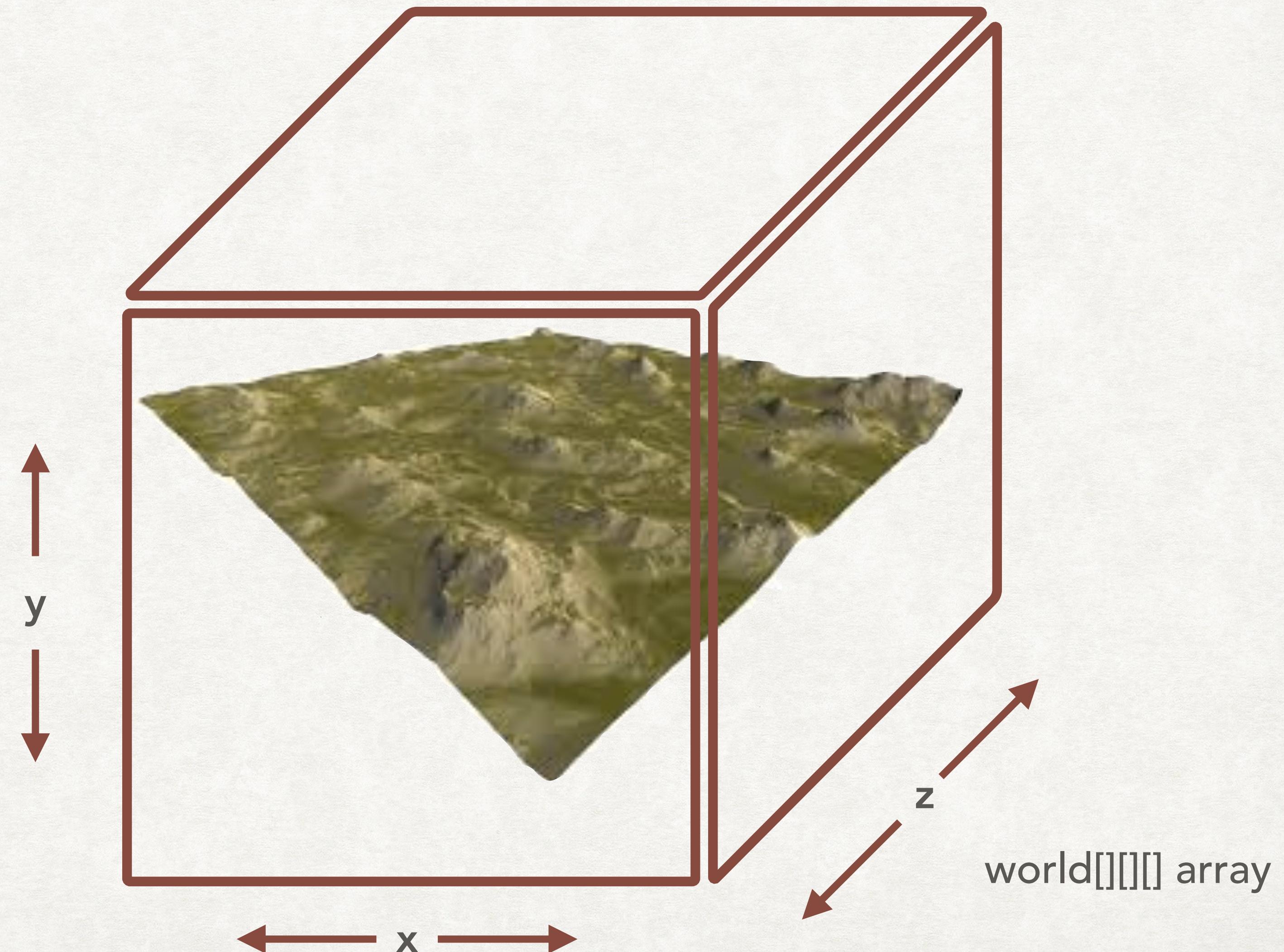
The Perlin noise function returns the y value for given x and z value.

```
y = perlin(x, z);
```

Use this in a pair of for loops to build the world.

```
for(x=0, x<100, x++)  
  for(z=0; z<100, Z++)  
    y = perlin(x, z);
```

Details on the parameters for Perlin noise are discussed in the Procedural Modelling lectures.



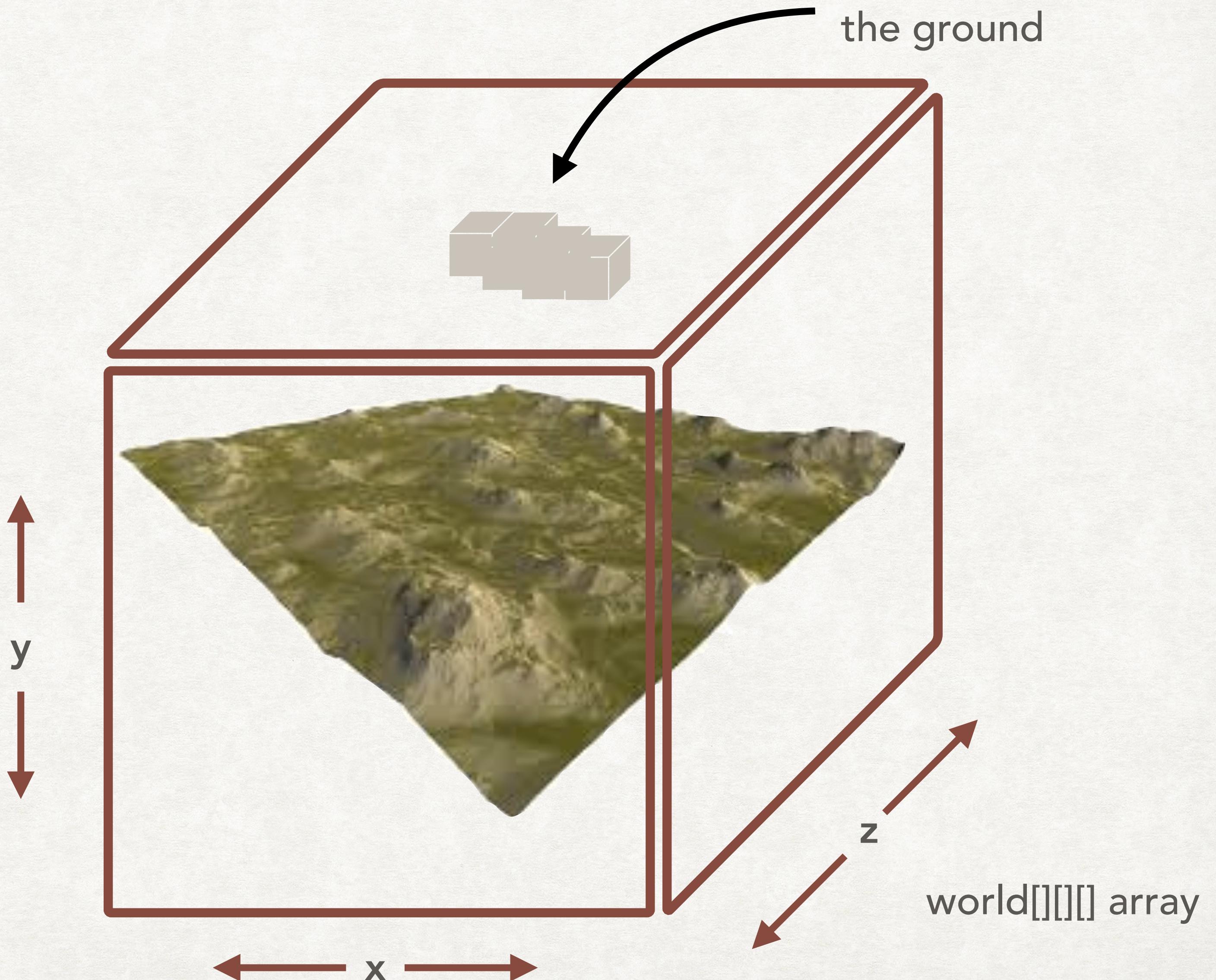
# ANIMATION

Create clouds above the procedural terrain. These will be white cubes in the `world[][][]` array.

Animate them by moving the location of the cubes in the array. The clouds should move across the sky.

You will need to store the location of the cloud cubes so it can be updated when they move.

Use the update function to move the clouds.

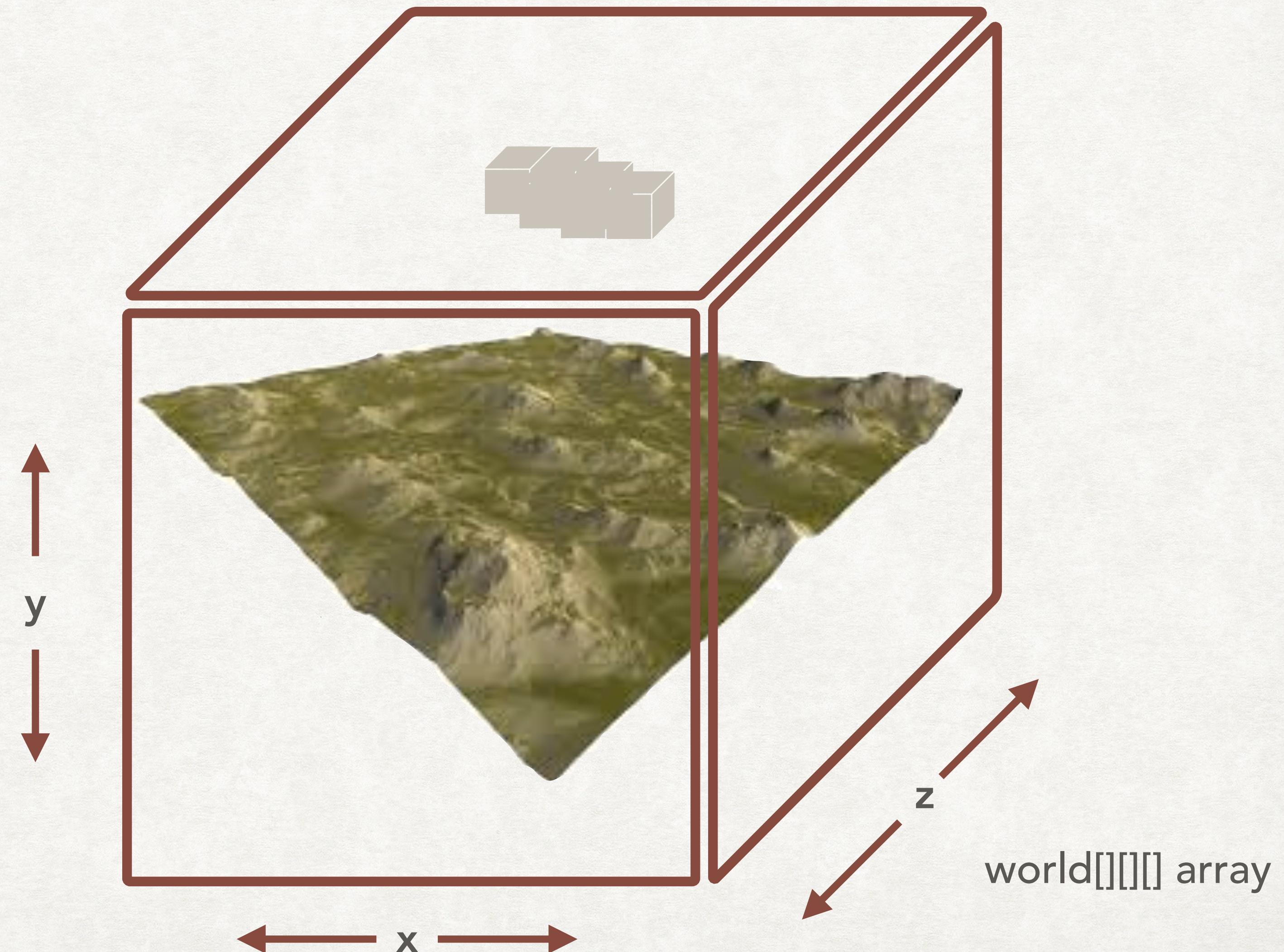


# PARAMETERS

A lot of this assignment requires good choices of parameters. These will affect the shape of the procedurally generated outdoor level and the speed of the animated clouds.

Pick parameters that make the game enjoyable. Don't make the outdoor level too uneven for the player to move across it.

Make the clouds move fast enough that they can be seen to move but not so fast that they are unrealistic.



`world[][][]` array

