Decorating the Terrain

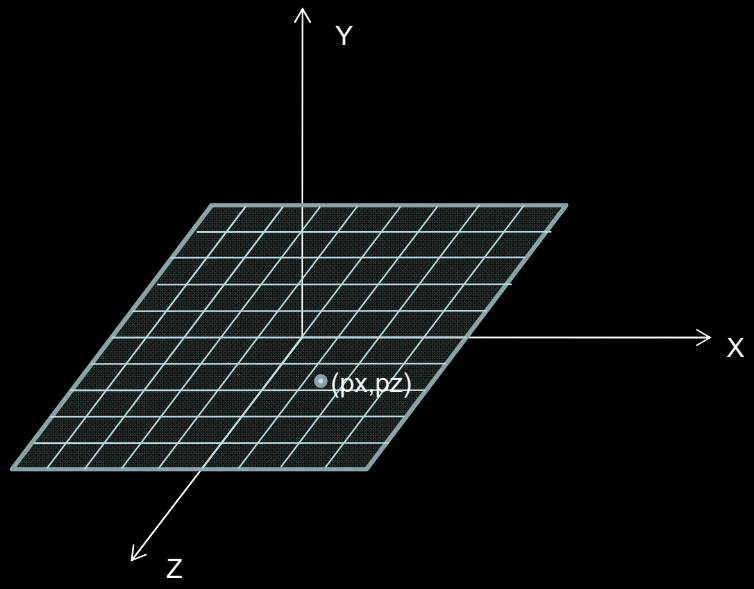
Height at any point in the terrain

- Problem: compute the height of point (px,pz) in the grid.

Considering `h(int x,int y)` as the function that returns the height at the vertices of the grid (previous assignment), we need to be able to compute the height for any point inside the grid.

The point (px, pz) is not a vertex on the grid, it is inside a grid cell.

We need a function `hf(float x, float z)`, where x, z are coordinates inside the grid cells.



Height at any point in the terrain

- Assume that the grid cells are square and unit length, and function `h(int x,int z)` provides access to vertex height values of a grid cell (yellow dots).

- The height at `(x1,pz)` can be obtained through linear interpolation of the heights at `(x1,z1)` and `(x1,z2)`. A similar process is used to compute the height at `(x2,pz)`. These are represented as green dots.

  - Let `fz` be the fraction part of `z`:

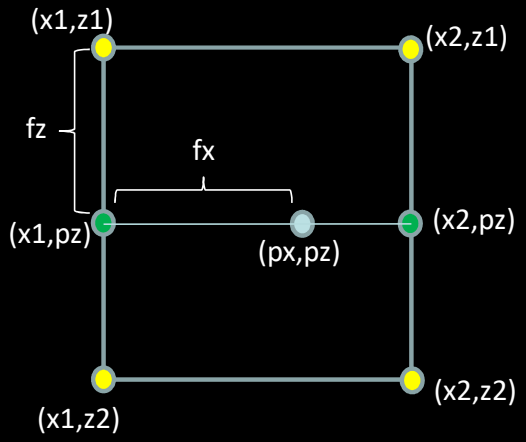
    - `fz = pz - z1;  // 0 <= fz <= 1`

  - `h_x1_z = h(x1,z1) * (1-fz) + h(x1,z2) * fz`
  - `h_x2_z = h(x2,z1) * (1-fz) + h(x2,z2) * fz`

- The height at `(px,pz)` (blue dot) is computed using linear interpolation between the heights for `(x1,pz)` e `(x2,pz)` (green dots)

  - `height_xz = h_x1_z * (1 - fx) + h_x2_z * fx`

```
x1 = floor(px); x2 = x1 + 1;
Z1 = floor(pz); z2 = z1 + 1;
```



Assignment

- Redo script for object placement on class 5, on top of the terrain from class 6.
  - First implement hf(x,z): function that provides the height at an arbitrary terrain point
  - Use hf(x,z) as the height at which items (trees, teapots, torus) are placed.