Code structure design

1. Level- each level holds collection of rooms in a specific layout.
   1. Tileset- dictionary keying certain character sequences in the mapcode to a surface tile. The tileSet will be generic (that is, in all cases the image of the floor tile will be keyed to ‘f’) so we’ll have to decide which tiles are used frequently enough to justify pre-loading for every level, and which tiles/images are unique enough to justify loading them separately on an as-needed basis
   2. Rooms- a list of all the rooms in the level. Will always have at least 2 pre-determined rooms: entry room and boos room. All other rooms are procedurally generated and appended to this list by the Room Generator.
   3. Room Generator
   4. ScrpitSet-
   5. SpriteSet-
2. Room- holds an image(surface) representing the visual layout of the room, the bounds of a room, plus any/all obstacles and (active) sprites.
   1. Surface[]
      1. Floor
      2. FloorCos
      3. Walls
      4. WallCos
      5. Blocks
      6. Sprites
   2. Spritelist
   3. Bounds
   4. obstacles
3. sprite