**a.**

**Generating pitch features (Grass Patterns)**

Cellular Automata’s controllability, randomness and visual coherence would be mostly based on initial generation. Each cell can only be in one of 2 states and rules are followed for the creation of each generation of cells. This would allow decorative patterns to be made quite easily and as you don’t need to worry about things like connectivity due to it just being decorative instead of being rooms etc. While there isn’t much designer control due to its ease of implementation you can just generate several examples until you find one you like. However, if you are planning for a specific look or theme this is unlikely to be the one you want to use (if you want images of footballs or something to appear that would be entirely due to randomness while other options can have this factored in)

Agent-based Generation would have greater controllability and visual coherence than CA’s as well as randomness (depending on how you set the 2 up). Agent based would provide greater control allowing for a slightly more tailored design, it however may be a bit more difficult to get a patterned design that a planned CA might give.

Constraint Satisfaction Problems compared to CAs is liable to have less chaos in its randomness so may help make for nicer/coherent patterns instead of completely random shapes/colors. Also CAs are based on cells having 2 states while CSPs can have a greater number (So if the patterns are colored CA would only produce a 2 colored pattern while CSP can give more variety) however the results ultimately come down to the constraints so if not careful then the resulting pattern may just be complete nonsense.

Wave Function Collapse would generate based on local constraints however this is more along the lines of matters like aspects of a dungeon (Floor, wall, door, etc) but in this case it is just patterns/colors. Like with CSPs a greater number of options are there but care would need to be taken when defining neighbors, however if done well interesting fractal patterns could be created. This can lead to quite a lot of variety but may be rather slow in generation depending on arrangement (is the pitch split into 6 rows? 10? Etc)

Ultimately, I would use Cellular Automata for the pitch features as it is just a nice decorative matter and not the main focus of the reason for being in the stadium. (It’s nice to have but not the focus of the event). You have no need to spend too much time or effort on this so making use of CAs ease of implementation. While it may not make the most expressive/impressive design for the audience, they are unlikely to think much on it unless it ends up looking remarkably odd/impressive.

**b.**

**Wave Function Collapse (terrain/environment) & Agent-based methods (objects/environmental entities)**

Wave function can be used on a larger scale than the agents to allow decent control over environment change while having greater low scale randomness so one cell in WFC could be an entire region while agents fill each region with the appropriate entities. However, both WFC and agent based methods have the issues of possibly creating dead-zones or areas without access. Depending on the rule or constraints you put in there is no guarantee that the overall map ends up playable and the addition of Agents wouldn’t prevent this from happening (possibly contributing to such issues).